

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"  
УДК 004.02

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис) О.В. Коваль  
(ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

## Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення  
за спеціалізацією Інженерія програмного забезпечення розподілених систем  
на тему “Підсистема інтеграції тривимірної моделі об’єкту в мобільну  
систему навігації ”

---

Виконав: студент 6 курсу, групи \_ТВ-81мп\_

Маруня Анастасія Василівна

(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник доц., к.т.н. Гагарін О. О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент

(підпис)

Київ — 2019

**Національний технічний університет України**  
**“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності — 121 Інженерія програмного забезпечення

за спеціалізацією — Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Коваль О.В. \_\_\_\_\_  
(прізвище, ініціали) (підпис)  
«\_\_\_\_» \_\_\_\_\_ 2019 р.

**З А В Д А Н Н Я**  
**НА МАГІСТЕРСКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Маруні Анастасії Василівни

(прізвище, ім'я, по батькові)

1. Тема дисертації «Підсистема інтеграції тривимірної моделі об'єкту в мобільну систему навігації»

Науковий керівник доц., к.т.н., Гагарін Олександр Олександрович  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “04” листопада 2019 року № 3812-с

2. Строк подання студентом дисертації 9 грудня 2019 р.

3. Об'єкт дослідження засоби створення та інтеграції тривимірної моделі об'єкту для системи навігації

4. Предмет дослідження Засоби створення та інтеграції схеми будівлі для потреб внутрішньої навігації в смартфонах

5. Перелік питань, які потрібно розробити  
Проаналізувати методи створення та інтеграції схеми будівлі. Розробити сервіс інтеграції схеми та перетворення в тривимірну модель об'єкту. Інтегрувати підсистему з іншими складовими частинами програмного продукту. Протестувати додаток.

6. Орієнтований перелік ілюстративного матеріалу «Актуальність теми дослідження», «Складові процесу інтеграції моделі тривимірного об'єкту», «Мета та завдання роботи», «Об'єкт, предмет дослідження», «Взаємодія модулів додатку», «Порівняння методів інтеграції схеми», «Запропонований підхід», «Метод SLAM», «Маркери ArUco», «Принцип визначення кватерніону», «Структура системи», «Алгоритм роботи з програмою», «Демонстрація роботи», «Висновки»

7. Орієнтований перелік публікацій \_\_\_\_\_  
Проблема формування схеми замкнутого простору у системах внутрішньої навігації  
(XVII міжнародна науково-практична конференція молодих вчених та студентів  
«Сучасні проблеми наукового забезпечення енергетики»)  
\_\_\_\_\_  
\_\_\_\_\_

8. Дата видачі завдання « 28 » вересня 2018 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. — 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 — 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 — 29.03.19 р.	
6	Розробка програмного продукту	03.06 — 25.10.19 р.	
5	Переддипломна практика	02.09 — 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 — 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	16.12.19 р.	

Студент

\_\_\_\_\_  
( підпис )

Маруня А. В.

\_\_\_\_\_  
(прізвище та ініціали)

Науковий керівник

\_\_\_\_\_  
( підпис )

Гагарін О. О.

\_\_\_\_\_  
(прізвище та ініціали)

# РЕФЕРАТ

## **Структура і обсяг дипломної роботи:**

Магістерська дисертація складається з 6 розділів, містить в собі 32 посилання на джерела за переліком, 22 ілюстрації та 23 таблиць та одного додатку. Обсяг роботи складає 99 сторінок.

## **Актуальність теми:**

З використанням сучасної техніки з'являється необхідність збільшення зручності навігації не лише від однієї адреси до іншої, а й всередині будівлі. Так як сфера навігації в приміщенні є не надто розвиненою і лише на етапі дослідження та розробки, а існуючі технології або не надають прийнятної точності, або потребують додаткового дорого обладнання.

Сучасні смартфони надають можливість проводити аналіз отриманих зображень та надання інформації, необхідної для отримання точної позиції та переміщення користувача, а також відображення карти будівлі. Так як смартфони на сьогоднішній день є майже у кожного – додаток є не лише актуальним, а й не вимагає складного встановлення чи проблем в користуванні, так як технології є безкоштовними.

**Метою дослідження** є створення підсистеми для системи навігації в будівлі з використанням смартфона, що буде надавати можливість відображення карти будівлі, вибору точку призначення та перегляду шляху.

## **Задачі дослідження:**

- розглянути, проаналізувати та порівняти існуючі методи побудови та відрисовування схеми карти на сцені додатку;
- розглянути та реалізувати можливість запису додаткової інформації (метаданих) в схему, а також реалізувати маркери для встановлення позиції користувача на карті;
- інтегрувати підсистему з іншими для отримання цілісної системи (додатку).

**Об'єктом дослідження** є система внутрішньої навігації з використанням доповненої реальності та комп'ютерного зору. Був виконаний аналіз існуючих шляхів

вирішення проблематики відрисовування схеми будівлі з використанням маркерів та метаданих, повороту карти та зроблений виносненок щодо переваг та недоліків. Створено модуль інтеграції тривимірної моделі об'єкту з використанням AutoCAD та маркерів ArcGIS.

**Предметом дослідження** є засоби інтеграції та відрисовування схеми, використання метаданих та маркерів в системі, засоби взаємодії користувача з картою будівлі.

**Методи дослідження:**

Для розв'язання поставлених задач використовується об'єктно-орієнтований підхід для цілісного аналізу предметної області, теорія використання кватерніонів для реалізації повороту карти, а також методи проектування інтеграції схематичної карти зображення на тривимірну модель будівлі.

**Наукова новизна:**

Удосконалено систему тривимірної інтеграції схеми для навігації по будівлі з використанням синтетичних маркерів, що надало можливість розробити систему навігації без використання додаткового дорогого обладнання. Також додаток при використанні безкоштовних технологій надає високу точність навігації.

Практичним значенням є те, що додаток може використовуватися не лише для навігації в навчальних закладах, а й для використання в інших галузях при доопрацюванні та заміні схеми будівлі.

**Апробація:**

Результати досліджень були представлені на XVII науково-практичній конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», що проходила в Києві 23-26 квітня 2019 року.

**Ключові слова:** НАВІГАЦІЯ, ДОПОВНЕНА РЕАЛЬНІСТЬ, ТРИВИМІРНА ІНТЕГРАЦІЯ, ВНУТРІШНЯ НАВІГАЦІЯ, МАРКЕРИ.

# ABSTRACT

## **The structure and volume of the graduate work:**

The master's thesis consists of 6 sections, contains 32 references to the sources in the list, 22 illustrations and 23 tables and one appendix. The workload is 99 pages.

## **Actuality of theme:**

Using modern technology combines the need to increase the ease of navigation not only from one address to another, but also inside the building. As the area of navigation in the room is not too advanced and only at the stage of research and development, and existing technologies either do not provide acceptable accuracy, or require additional expensive equipment.

Modern smartphones provide an opportunity to analyze the images obtained and provide the information needed to get the exact position and movement of the user, as well as display a map of the building. Since almost everyone has smartphones today, the app is not only up-to-date, but also requires no complicated installation or problems, as the technologies are free.

**The purpose of the study** is to create a subsystem for the navigation system in the building using a smartphone, which will be able to display the map of the building, select a destination and view the path.

## **Research objectives:**

- review, analyze and compare existing methods of constructing and drawing the map scheme on the stage of the application;
- consider and implement the ability to write additional information (metadata) in the scheme, as well as implement markers to establish the position of the user on the map;
- integrate the subsystem with others to obtain a coherent system (application).

**The object of the study** is an internal navigation system using augmented reality and computer vision. An analysis of the existing ways of solving the problem of drawing the scheme of the building with the use of markers and metadata, rotation of the map was carried out and the advantages and disadvantages were made. A module for integrating a 3D object model using AutoCAD and ArUco tokens has been created.

**The subject of the research** are the means of integration and drawing of the scheme, the use of metadata and markers in the system, the means of user interaction with the building map.

**Research methods:**

To solve these problems, we use an object-oriented approach for holistic analysis of the subject area, the theory of the use of quaternions to realize the rotation of the map, as well as methods of designing the integration of a schematic image map on a three-dimensional model of the building.

**Scientific novelty:**

Improved three-dimensional system integration scheme for navigation on the building using synthetic markers, which made it possible to develop a navigation system without the use of expensive expensive equipment. Also, when using free technologies, the application provides high accuracy of navigation. The practical value is that the application can be used not only for navigation in educational institutions, but also for use in other industries when refining and replacing the building.

**Approbation:**

The results of the research were presented at the 18th Scientific and Practical Conference of Young Scientists and Students "Modern Problems of Scientific Support of Energy", which took place in Kyiv, April 23-26, 2019.

**Keywords:** NAVIGATION, COMPLETED REALITY, THREE-DIMENSIONAL INTEGRATION, INTERNAL NAVIGATION, MARKERS

## ЗМІСТ

Перелік скорочень, умовних позначень та термінів .....	10
Вступ .....	11
1. Задача розробки підсистеми інтеграції тривимірної моделі об'єкту в мобільну систему навігації.....	13
1.1 Задача відображення схеми в додатку .....	13
1.2 Призначення програмного продукту .....	15
1.3 Вхідні дані .....	19
1.4 Компоненти системи .....	19
Висновки до розділу 1.....	20
2 Аналіз проблеми створення підсистеми інтеграції тривимірної моделі об'єкту в мобільну систему навігації.....	22
2.1 Існуючі рішення .....	22
2.2 Порівняння існуючих систем .....	26
2.2.1 INPLACES .....	27
2.2.2 Hubbell IPS .....	28
2.2.3 IndoorAtlas .....	29
Висновки до розділу 2.....	31
3 Опис побудованої моделі та програмної реалізації .....	33
3.1 Бібліотека netDXF.....	33
3.2 Запис метаданих з використанням XDATA AutoCAD .....	37
3.3 Маркери ArUCO.....	40
3.4 Реалізація повороту карти на сцені додатку .....	46
3.4.1 Визначення орієнтації.....	46
3.4.2 Поняття кватерніону .....	50



3.4.3 Реалізація схеми та перетворення в тривимірну карту .....	52
Висновки до розділу 3.....	57
4 Засоби розробки.....	59
4.1 Ігровий рушій Unity.....	59
4.2 AutoCAD .....	61
4.3 Інтегроване середовище розробки Microsoft Visual Studio .....	62
4.4 Мова програмування C#.....	63
4.5 Платформа ARCore.....	63
Висновки до розділу 4.....	66
5 Робота користувача з системою.....	67
Висновки до розділу 5.....	70
6 Розробка стартап проекту.....	71
6.1 Опис ідеї проекту.....	71
6.2 Технологічний аудит ідеї проекту .....	73
6.3 Аналіз ринкових можливостей запуску стартап-проекту .....	74
6.4 Розробка ринкової стратегії проекту .....	82
6.5 Розроблення маркетингової програми.....	85
Висновки до розділу 6.....	90
Висновки .....	91
Список використаних джерел .....	93
Додаток А.....	96

## **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ**

GPS — Global Positioning System — система глобального позиціонування.

QR-код — QR code — код швидкої відповіді.

SLAM — Simultaneous Location And Mapping — одночасна локалізація та картографування.

SfM — Structure from Motion — метод відновлення структури з руху.

BLE — Bluetooth Low Energy — специфікація бездротової технології обміну даними з низьким споживанням електроенергії.

## ВСТУП

Під час розробки модуля підсистеми інтеграції тривимірної моделі об'єкту в мобільну систему навігації постає питання не лише вигляду вхідних даних, а й побудова та відрисовування схеми з необхідною точністю для подальшого використання схеми іншими модулями програми.

Зазвичай, схеми будівель мають вигляд побудованої вручну схеми на папері в певному масштабі. В кращому випадку схема буде перенесена в електронний вигляд за допомогою програм AutoCAD чи Kompas. [1]

Звичайно, є безліч сучасних методів отримання даних та побудови схеми без використання сторонніх програм та перенесення даних в електронний вигляд вручну. Одним з найбільш широко використовуваних методів є SLAM (Simultaneous localization and mapping). [2] Даний метод виокремлює характерні точки з отриманого потоку зображення та записує їх в хмару. При повторному проходженні маршруту алгоритм вирізняє та порівнює характерні точки з уже отриманими раніше. Але подібний метод має низку недоліків. Адже виокремлені точки можуть бути на предметі, що може бути зсунутим і вже карта не матиме потрібної точності.

Існуючі рішення не надають потрібної точності навігації в будівлі, так як не мають початкові дані, вони збираються при використанні додатків, а також використовують додаткове обладнання. Саме тому було розглянуто використання попередньо сформованої схеми будівлі, що містить в собі ідентифікатори для знаходження позиції користувача, а також надає можливість взаємодіяти з користувачем в тривимірному просторі.

Для використовуваного алгоритму локалізації SLAM необхідна точна схема, яка матиме набір точок для порівняння зі зчитуваними даними. Схема має бути побудована не лише з заданою точністю для коректної побудови маршруту в додатку навігації, а також містити такі дані, як точна довжина стіни, місцезнаходження

дверей, а також ряд метаданих, що будуть відрізняти аудиторії одна від одної, а також такі об'єкти як сходи та їх напрямки.

Основною проблемою роботи є створення підсистеми тривимірної інтеграції об'єкту в мобільну підсистему навігації, використовуючи за вхідні дані попередньо створену двовимірну схему. Схема має містити дані, мітки та маркери, що будуть використані для взаємодії з іншими підсистемами додатку, таких як підсистема позиціонування та підсистема знаходження і побудови шляху.

Для інтеграції двовимірної схеми та перетворення її на тривимірну з можливості взаємодії та масштабування карти користувачем, було розглянуто можливість використання кватерніонів, що надає можливість маніпулювати з розміром карти та кутом її повороту. Також це необхідно, щоб карта переміщалась разом з користувачем при проходженні карти.

Перший розділ роботи розглядає задачу, що вирішується. В другому розділі задача аналізується, розглядаються існуючі рішення, а також надається оцінка з запропонованим рішенням. Третій розділ складається з використаних технологій розробки підсистеми, алгоритми роботи та їх застосування. Четвертий розділ оглядає використані засоби розробки. Передостанній, п'ятий, розділ присвячений роботі користувача та підготовці схеми для інтеграції в додаток. Останній розділ оцінює конкурентоспроможність запропонованого додатку та можливість виведення на ринок.

# **1. ЗАДАЧА РОЗРОБКИ ПІДСИСТЕМИ ІНТЕГРАЦІЇ ТРИВИМІРНОЇ МОДЕЛІ ОБ'ЄКТУ В МОБІЛЬНУ СИСТЕМУ НАВІГАЦІЇ**

Для розгляду задачі створення підсистеми інтеграції тривимірної моделі об'єкту в мобільну систему навігації необхідне розуміння, що розуміється під тривимірною моделлю об'єкту та який базовий стан вона приймає. Маються на увазі вхідні дані, що будуть використані для створення тривимірної моделі та що саме для цього потрібно.

## **1.1 Задача відображення схеми в додатку**

Для коректного відпрацювання інших модулів системи необхідно, щоб були задоволені такі умови як точність схеми, формат в якому вона буде завантажуватися та які об'єкти будуть відрисовуватися.

Схема (грец. Σχῆμα — образ, вид) — спрощений вид чи зображення чогось. Це може бути прилад, машина, будівля чи функція, що передає основну ідею чи принцип, використовуючи умовні позначення, приписи та метадані для надання характеристики, за допомогою якої об'єкти схеми будуть визначатися у відповідності заданих умов.

Звісно, на сьогоднішній день можна успішно побудувати карту місцевості та перевести її в схематичний вигляд лише за допомогою супутникових знімків. Для цього необхідний лише GPS-навігатор, ввімкнене відслідковування точки геокоординат та підключення до мережі інтернет.

GPS - це пристрій, здатний отримувати інформацію від супутників GPS, а потім обчислювати географічне положення пристрою. Використовуючи відповідне програмне забезпечення, пристрій може відображати положення на карті, і воно може пропонувати вказівки. Система глобального позиціонування (GPS) - це глобальна

супутникова навігаційна система (GNSS), що складається з мережі мінімум 24, але наразі 30 супутників, розміщених на орбіті Міністерством оборони США. [3] GPS спочатку був розроблений для використання військовими Сполучених Штатів, але в 1980-х роках уряд Сполучених Штатів дозволив використовувати систему в цивільних цілях. Хоча дані супутника GPS безкоштовні і працюють у будь-якій точці світу, GPS-пристрій та пов'язане з ним програмне забезпечення потрібно купувати чи орендувати. Пристрій GPS може отримувати інформацію про місцезнаходження та час у системі GPS за будь-яких погодних умов, у будь-якій точці Землі або поблизу.

Приймання GPS вимагає безперешкодної прямої видимості до чотирьох або більше GPS-супутників і піддається поганим умовам супутникового сигналу. У винятково поганих умовах сигналу, наприклад, у міських районах, супутникові сигнали можуть демонструвати багатостороннє поширення, коли сигнали відштовхуються від структур або ослаблені метеорологічними умовами. Заборонені лінії зору можуть виникати на балдахіні дерева або всередині споруди, наприклад у будівлі, гаражі чи тунелі. Сьогодні більшість автономних GPS-приймачів використовуються в автомобілях. Можливість GPS-смартфонів може використовувати технологію допоміжної GPS (A-GPS), яка може використовувати базову станцію або вежі стільникового зв'язку, щоб забезпечити швидший час до першого виправлення (TTFF), особливо коли сигнали GPS погані або відсутні. Однак частина мобільної мережі за технологією A-GPS була б недоступною, коли смартфон знаходиться поза діапазоном мобільної приймальної мережі, тоді як аспект GPS інакше надалі буде доступний.

Але постає проблема, наявні супутникові знімки лише відкритої місцевості, але докладну схему будівлі по фото, що має вид даху побудувати не можливо. Тобто, у будівлі GPS-пристрій не буде надавати можливість навігації.

## **1.2 Призначення програмного продукту**

Метою розробки є створення програмного продукту, що є підсистемою програми навігації навчальним корпусом; він надаватиме додатку необхідну схему будівлі для навігації та отримання точного місцеположення користувача, точки з метаданими аудиторій для прокладення маршруту та вибору кінцевої точки маршруту в додатку, а також забезпечуватиме користувача зручним інтерфейсом, що полегшить користування програмним продуктом.

В модулі мають поєднуватися не лише схема, а й набір відмічених маркерів. Схема має бути побудована з точністю до 20 см для коректної роботи додатку та відображення маршруту. Також, при генерації схеми на сцені додатку мають відрисовуватися лише стіни схеми без об'єктів, що містять метадані.

Отже, програмний продукт повинен:

- давати можливість користувачу обирати кінцеву точку призначення використовуючи записані в схемі метадані (з випадającego списку);
- надавати можливість встановлювати кінцеву точку на карті вручну (користувацький режим);
- відрисовувати маршрут, враховуючи об'єкти на схемі;
- відрисовування на сцені об'єктів з використанням доповненої реальності;
- забезпечувати перегляд всієї карти, по якій здійснюється навігація;
- підлаштовуватися під різну роздільну здатність екрану;
- надавати можливість перегляду схеми;
- дозволяти змінювати режим перегляду (режим карти, режим камери)
- мати зрозумілий всім користувачам інтерфейс.

Можливості додатку наведені на діаграмі прецедентів (рисунок 1.1)

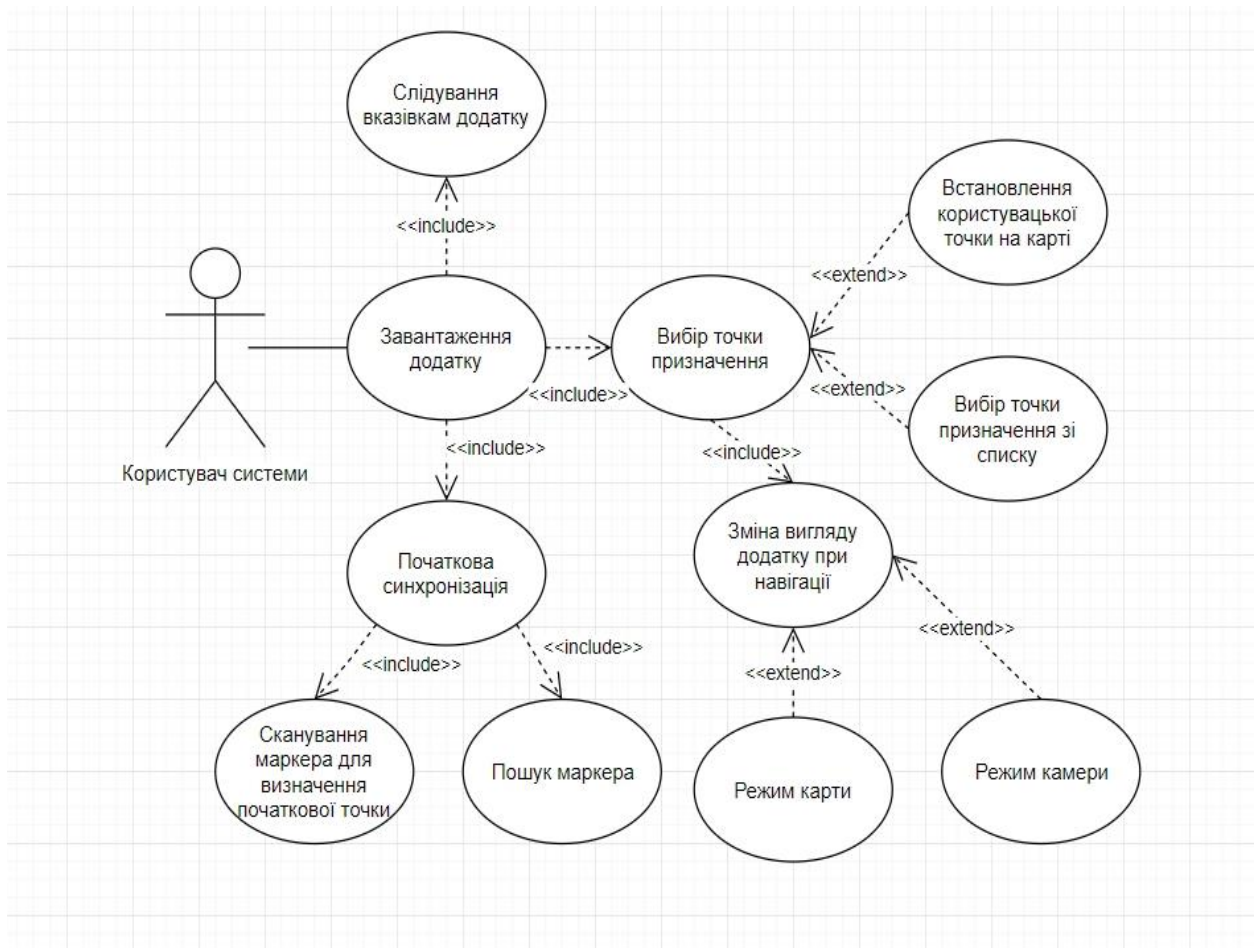


Рисунок 1.1 – Діаграма прецедентів додатку

Користувацьким сценарієм комплексного додатку – це навігація в приміщенні за допомогою запропонованих функціональних можливостей, таких як вибір точки призначення, побудови найкоротшого маршруту та пересування по карті в реальному часі.

- Розроблювана підсистема повинна задовольняти такі умови:
- розпізнавати об’єкти на завантаженій схемі;
- відрисовувати об’єкти у відповідності до його типу;
- розпізнавати положення маркеру на схемі у відповідності до кута дуги;
- розпізнавати метадані, що записані в об’єкти схеми;
- завантажувати метадані в випадуючий список для надання вибору кінцевої точки призначення для побудови найкоротшого маршруту;



- надавати можливість використання додатку без додаткового обладнання чи програм.

Додаток також має потенціал використання не лише в навчальних закладах, а й в інших сферах застосування, наприклад :

- сервіси для торгових та розважальних центрів з внесенням додаткових метаданих також можливо використовувати застосунок в показі робочих годин магазину, асортименту, а також внесенням контекстної реклами. Звичайно, в такому випадку додаток буде потребувати доопрацювання, щоб зберігати метадані в базі даних для подальшого використання. Також його можна пристосувати для пошуку необхідної точки. Якщо торговий центр має велику площу — доцільніше буде використовувати подібний додаток для відслідковування поточної позиції на карті, а також знаходження необхідної точки призначення за категорією, навіть між поверхами;
- сервіси відслідковування позиції на схемі. При можливій аутентифікації користувача за допомогою додатку, можливе визначення позиції іншого довіреного користувача у відповідності місцезнаходження в будівлі. Тобто, можливим буде відслідковування точної позиції людини в будівлі, наприклад, дитини, що знаходиться в розважальному відділі;
- в медицині для знаходження потерпілого і надання необхідної допомоги, якщо людині стало погано у незнайомому приміщенні і вона не може пояснити де саме знаходиться. В таких випадках є дуже важливе збереження часу, що може зберегти людині життя, тому в такому випадку пошук найкоротшого шляху є максимально необхідним;
- в військових сферах. Дуже частими випадками при веденні військових операцій існує проблема навігації по будівлі, а також розпізнавання «свій»-«чужий». При застосуванні додатку з необхідним доопрацюванням на карті буде можливість побачити місцезнаходження інших учасників операції, навіть якщо буде погана видимість. Необхідним є лише пристрій для роботи додатку та підключення до мережі Інтернет;

- транспортна навігація з використанням дистанційного керування чи автопілоту в будівлях. Це можуть бути як просто навігація користувача по паркінговій зоні, що являється приміщенням. Тобто для знаходження вільного місця на парковці чи для пошуку найбільш зручного виїзду. Також дуже частою є проблема, коли на великій за площею парковці складно знайти залишений там автомобіль. Систему можна адаптувати для збереження початкової точки чи попереднього маршруту. Також можна використати і для доставлення необхідних речей роботизованими кур'єрами. В великих офісних будівлях є частою необхідністю перенесення речей чи документів, наприклад, в архів або іншому співробітнику. Подібну операцію можна виконати для встановлення маршруту роботизованому кур'єрові;
- в навчальних та пізнавальних цілей. Наприклад, для прокладення екскурсійних маршрутів по музеях з використанням попередніх пояснювальних записів. Дуже частою є практика, коли біля експонатів є пояснювальні записи, що можна прослухати в навушниках. Але це є доволі негігієнічним, також програвач може бути не завжди вільним. В такому випадку є дуже зручним завантажити додаток, що не лише прокладе маршрут між експонатами, а також буде використовувати записані до точок метадані, щоб надавати необхідну інформацію;
- в логістиці. На великих складах є повністю автоматизована система навігації для переміщення, завантаження та отримання вантажу. Але для цього потрібне додаткове інфраструктурне обладнання, що є доволі дорогим. Це потрібно для збільшення точності навігації та дозволяє складській техніці під час перевезення з великою точністю розмістити вантаж у необхідному відділі збереження. Але для цього потрібні додаткові витрати, розміщення датчиків, а також їх налаштування.

Метою додатку є саме навігація користувача до необхідного місця призначення. Це може бути вихід з будівлі, аудиторія чи викладацький кабінет. Звичайно, додаток можливо доопрацювати для подальшого комерційного чи державного користування,

при необхідності. Тобто додаток можливо модернізувати у відповідності до вимог замовника, при цьому не вносити критичні зміни до його структури.

### **1.3 Вхідні дані**

Вхідними даними є базова схема з розмірами та масштабом, яку необхідно перенести в електронний вид необхідного масштабу. Також, метадані, що вносяться на схему на відповідні з реальними об'єкти.

Після цього ми отримуємо схему з усіма необхідними вхідними даними для завантаження в підсистему

### **1.4 Компоненти системи**

Для коректної роботи додатку необхідна синхронізація трьох підсистем :

Підсистема інтенрації тривимірної модклі будівлі, що розглядається в даній дипломній дисертації;

Підсистема позиціонування, що встановлює почицію користувача відносно схеми будівлі;

Та підсистема знаходження найкоротшого шляху, що надає можливість користувачеві обрати точку призначення та пройти найкоротшим маршрутом від точки А до точки Б.

Взаємодія компонентів системи зображена на рисунку 1.2.

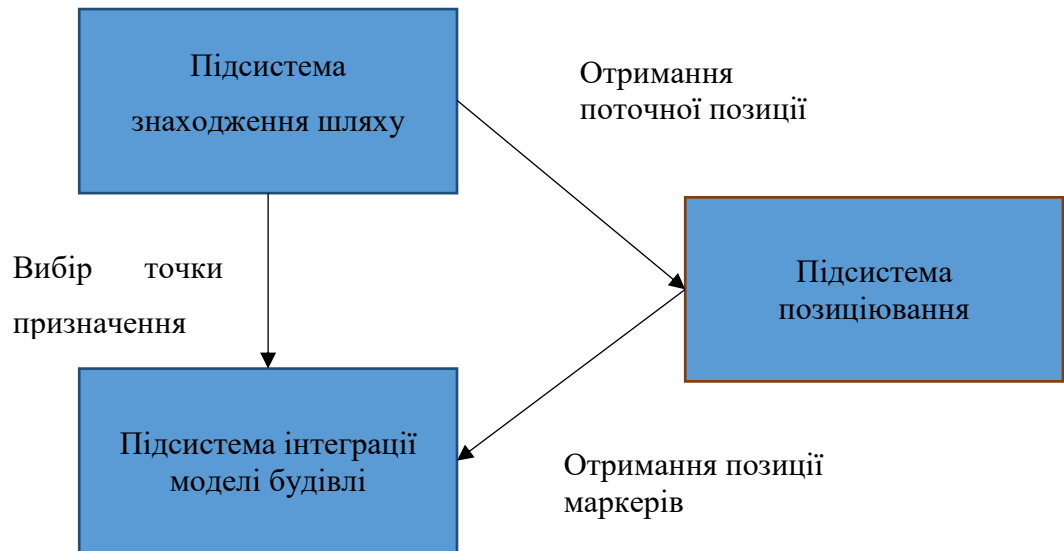


Рисунок 1.2 – Компоненти системи

Без коректної взаємодії хоча б одного модуля системи – додаток не зможе коректно прокласти маршрут користувачеві, тож дуже важливо, щоб вхідні дані оброблювалися у відповідній черзі. Спершу повинна бути завантажена схема з метаданими та маркерами та відрисовуватися на сцені додатку. Після чого вже стає можливим знаходження позиції користувача відносно схеми.

Коли вже встановлена позиція користувача, її можна задати як початкову точку та обрати кінцеву. Додаток прорисує найкоротший шлях відносно карти та передє переміщати курсор користувача відносно відрисованої схеми будівлі.

## Висновки до розділу 1

Таким чином, у розділі було розглянуто та обґрунтовано актуальність системи для навігації по будівлі. Є безліч існуючих рішень, що не задовольняють точність для навігації користувача за рахунок накопичення похибок та неможливості використання GPS-технологій.

Розглянуто сфери розвитку та використання додатку, що може застосовуватися не лише для навігації користувача, а й для використання додатку в логістиці, інтеграції та використання додатку роботизованої техніки.

Була розглянута структура системи, які ресурси повинні використовуватися та в якому порядку має оброблюватися інформація у відповідності до роботи системи.

В розділі поставлені задачі, які має вирішувати система під час роботи, а також можливості користувача під час використання додатку.

## **2 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ ПІДСИСТЕМИ ІНТЕГРАЦІЇ ТРИВИМІРНОЇ МОДЕЛІ ОБ'ЄКТУ В МОБІЛЬНУ СИСТЕМУ НАВІГАЦІЇ**

При розробці системи внутрішньої навігації замкнутого простору (приміщення) постає проблема оцінки універсальності системи, тобто чи зможе система адаптуватися, якщо буде використана в іншому замкнутому просторі.

В звичайних додатках навігації по місцевості використовується технологія GPS, що є доволі точною, але лише на відкритому просторі. В будівлі технологія GPS надає значні спотворення позиціонування і є доволі неефективною за рахунок зменшення доступності GPS сигналів за рахунок стін.

Було розглянуто декілька методів для визначення найбільш зручного за використанням, впровадженням та ціновою політикою.

### **2.1 Існуючі рішення**

Система позиціонування в приміщенні, яка базується на BLE (Bluetooth Low Energy), реалізується за допомогою маяків. [4] Маяком можна вважати невеликий електронний пристрій, що складається з мікросхеми антени чи інших датчиків на невеликого розміру схемі. Тобто, він є лише радіопередавачем, що надсилає сигнал.

Цей сигнал ловлять пристрої, які обладнані для його прийому (наприклад, смартфонів). Цей метод не використовує значних ресурсів смартфона, є доволі економним, а встановлення передавача доволі легке. Але точність не значна, всього 1-3 метри, що не задовольняє нашим умовам, а сам передавач є дорогим. Wi-Fi можна використовувати подібно до технології з використанням BLE, але він зовнішнього джерела живлення, додаткових витрат додаткове встановлення обладнання. В порівнянні з BLE, сигнал значно сильніший, і технологія працює на значно більші відстані.

Технологія з використанням NFC (Near Field Communication) -- це пристрій, що складається з невеликих мікросхем, які не потребують джерела живлення. Користувач смартфоном чи іншим пристроєм визначає цей чіп і зчитує серійний номер, для цього потрібно знаходитися не далі, ніж за 30 см від чіпа. [5]

Ця технологія може використовуватися для точного позиціонування в приміщенні в межах 30 см. Технологія доволі зручна і також відрізняється швидкістю встановлення з'єднання. Проте ця технологія NFC доступна тільки на нових телефонах, тому для її широкого використання потрібен деякий час.

Існує також технологія Structure from Motion, що теж використовується для систем внутрішньої навігації. Structure from Motion – це техніка комп'ютерного зору для побудови 3D-моделі, використовуючи попередньо зібрану базу фотографій приміщення.

Значно SfM відрізняється від інших методів позиціонування тим, що з комбінації попередньо створеної записаної хмари з характерними точками та порівнянням їх з розпізнаними точками на фотозображенні використанням орієнтації камери в просторі на момент зйомки можливо дізнатися позицію, що можуть бути використані для вирішення задачі навігації. [6] Загальна послідовність виокремлення тривимірних точок з набору фотографій є наступною :

1. Вилучення ознак із зображення.
2. Знаходження однакових точок за допомогою порівнянь зображень.
3. Перевірка орієнтації камери на момент зйомки еталонного фото та орієнтації на момент використання додатку.
4. Ущільнення хмари точок та створення полігональної моделі.

Реконструювання сцени починається з виділення найбільш підходящої пари зображень з урахуванням перерахованих пунктів. Потім до послідовно порівнюються та додаються інші зображення. Деякі з них можуть бути відкинуті з процесу, якщо їм неможливо знайти відповідні комплементарні фотографії. Але даний метод також має низку недоліків. Для задовільної точності необхідна велика кількість зображень та обчислювальних ресурсів для забезпечення прийнятної точності та складність отримання множини «правильних» фотографій зі значним перекриттям об'єктів.

Як наслідок, на поточний момент, метод не можна застосовувати для дуже великих просторових об'єктів. Перевагами ж є широкий вибір алгоритмів та можливостей їх конфігурації, наявні оптимізовані реалізації алгоритмів на графічних прискорювачах, достатня точність локалізації на невеликих наборах даних.

Існують різні способи вирішення проблеми навігації. Одним з них є навігація по відеоорієнтирам, в основі якої лежать методи комп'ютерного зору, але ці методи добре працюють лише в середовищах, параметри яких значно не змінюються (наприклад, освітлення), в складних же умовах робота даних методів є складною та вимагає перенастроювання.

Для вирішення даних проблем може застосовуватися метод штучних нейронних мереж, побудованих за принципами організації біологічних нейронних мереж - мереж нервових клітин живого організму. [7]

Перевагою методу нейронних мереж в управлінні є те, що він дозволяє синтезувати закон управління за допомогою однієї лише вибірки навчальної інформації. Вибірка містить вхідні дані (це можуть бути зображення з камери) і відповідні їм вихідні дані. Вихідні дані специфічні для кожної конкретної задачі, наприклад, в задачах розпізнавання образів на зображеннях виходом може бути інформація про те, який образ присутній (або відсутній) на кожному зображенні. Але знову ж постає проблема розпізнавання в навігації по будівлі.

Нейронна мережа може давати збій при розпізнаванні, наприклад, дверей кабінету, що не мають явних ознак, а також, якщо освітленні зміниться. Для цього потрібно постійно донавчати нейронну мережу, що не підходить для мобільного додатку. Вирішенням даної проблеми є застосування методу SLAM (Simultaneous Localization And Mapping) – це метод одночасної навігації і побудови карти. Метод використовується роботами чи іншими засобами пересування, що використовують автономний режим, потрібна лише камера для побудови карти на невідомій місцевості чи приміщенні або поновленні у відповідності до змін вже існуючої карти, використовуючи як точки характерні об'єкти чи риси. Якщо розпізнавання в приміщенні – це великі предмети, кути, картини, вікна, двері тощо.



Використання маркерів спрощує метод – алгоритм відштовхується від конкретних встановлених на попередньо завантаженої схемі будівлі. Це може бути датчик чи більш простий варіант, що був використаний для додатку QR-код. Тобто, додаток при зчитуванні коду визначає місцеположення на карті і вже за допомогою алгоритму SLAM триангулюється відстань, що задана в масштабі на схемі. В результаті ви можете обчислити точне співвідношення між початковою точкою (QR-кодом) та власним розташуванням. Якщо вам потрібно переміститися – ви можете зробити висновок, куди і як потрібно рухатися.

В даному випадку при вже наявній карті місцевості алгоритм дозволяє з високою точністю побудувати маршрут до наступного потрібного маркера. Тобто, найкоротший маршрут з точки А в точку Б і не потребує складного обладнання, донавання чи великої бази знань.

Вирішенням проблеми може бути застосування методу SLAM (Simultaneous Localization And Mapping) – це метод одночасної навігації і побудови карти-схеми. Метод використовується роботами чи автономними засобами пересування, потребують лише набір камер для побудови карти-схеми на невідомій місцевості чи приміщенні або оновленні вже існуючої схеми, використовуючи методи виокремлення типових точок з характерними рисами. Якщо розпізнавання виконується в приміщенні ці точки є кути, вивіски, написи, предмети, що мають характерні риси. Якщо ж на відкритій місцевості, наприклад, в полі – це можуть бути кущі, що знаходяться на певній відстані один від одного.

Метод SLAM є концепцією, що поєднує в собі два процеси, що не залежать один від одного, в цикл послідовних розрахунків. Концепцію методу можна пояснити на прикладі лабіринту та робота: у нас є невідомий лабіринт, в якому знаходиться робот, що має лише камеру і йому необхідно за допомогою здвигів зображення на карті побудувати повну карту лабіринту для того, щоб наступний, хто потрапить в лабіринт не блукав, а вже чітко міг побудувати маршрут виходу з лабіринту. На практиці це означає, що робот шукає відповідь на запитання «Де я?» і «Як виглядає світ навкруги?».

Метод SLAM можна описати як постійно повторювана послідовність кроків:

- сканування навколишнього середовища;
- визначення зміщення на основі поточного кадру з попереднім;
- виділення на поточному кадрі характерних міток і відображення їх як точок;
- зіставлення точок кадру з мітками, що були отримані при попередньому скануванні приміщення;
- оновлення на основі отриманої інформації положення на карті;
- перевірка на цикл – чи не проходимо ми повторно по тій же місцевості.

Такий алгоритм дозволяє формувати не лише карту місцевості, а й траєкторію переміщення, яку можна використовувати для побудови наступного маршруту.

Постійне коригування карти при повторному проходженні необхідно для зменшення похибки визначених характерних точок. Тобто при першому проходженні мітки можуть бути встановлені на об'єкти, що рухаються і при наступному проходженні їх вже немає на мапі і їх можна відсіяти і завдяки чому залишити лише нерухомі мітки. Кожному виділеному об'єкту зіставляється опис, що дозволяє його ідентифікувати на кожному кадрі, на якому він існує.

Організувати процес побудови схеми для звичайного користувача нескладно достатньо лише камери телефону та організації обробки отриманих знімків методом SLAM в режимі реального часу. При цьому можливо відслідковувати динамічні (рухаючі) об'єкти при кожному наступному проходженні маршруту.

## **2.2 Порівняння існуючих систем**

На сьогоднішній день питання позиціонування та навігації досить широко розглядається в новітніх технологіях. Це не лише навігація користувача, а й навігація роботизованої техніки. Розглянемо та порівняємо існуючі системи.

### **2.2.1 INPLACES**

Області застосування послуг з розширеною реальністю на основі локації є безліч! Наприклад, навігація в приміщенні ідеально підходить як мобільний путівник на ярмарках та конференціях, а також у громадських будівлях .

В умовах промислових процесів доповнена реальність на основі місця дає змогу повторно переживати свої дані та розкриває нові аспекти ваших процесів.

Функції:

- система позиціонування в приміщенні. Ті, хто хоче запропонувати послуги на основі місцезнаходження, повинні ознайомитися з місцем розташування;
- дозволяє ознайомитись з місцем розташування та надає дані про позиціонування для служб розширеної реальності на основі місцезнаходження;
- використовує нові технології стандартних смартфонів, таких як ARKit та ARCore, і, таким чином, пропонує високий ступінь масштабування; [8]
- суто базується на даних і не залежить від апаратних інфраструктур, таких як маякові або світлодіодні системи;
- має позиціонування з точністю до сантиметра та забезпечує плавне, безшовне відстеження, подібне до досвіду зовнішньої навігації за допомогою GPS.

Впливи:

- послуги з додатковою реальністю на основі місцезнаходження. Ті, хто знає місцезнаходження, можуть запропонувати послуги на основі місцезнаходження;
- пропонує інноваційний користувацький інтерфейс, за допомогою якого ваша послуга на основі розташування відбувається у занурювальному 3D-світі в режимі реального часу.

INPLACES - це компонент програми, який може бути інтегрований як функція в будь-яку службу на основі локації, наприклад, навігацію в приміщенні, презентацію продуктів або розгортання даних/процесів. INPLACES інтегрується з двомірним

світом карт Google, забезпечуючи безперебійну навігацію між внутрішнім та відкритим простором.

### **2.2.2 Hubbell IPS**

Великі заклади, такі як магазини роздрібної торгівлі, лікарні, аеропорти, місця проведення заходів та інші складні середовища, можуть відчувати себе незнайомими та стати дезорієнтуючими для мешканців. Це часто призводить до того, що мешканці не можуть легко зорієнтуватися в приміщенні та орієнтуватися на певну визначну точку. Система позиціонування в приміщенні Hubbell (IPS) робить гіперточне розташування в реальному часі можливим у внутрішніх комерційних та промислових програмах, використовуючи спеціалізовану технологію, що легко вбудовується в світильники, що надаються Hubbell та сторонніми партнерами. [9]

Система позиціонування в приміщенні Hubbell (IPS) використовує світильники, оснащені технологією Encoded Light, також відомою як Visible Light Communication (VLC) і технологією Bluetooth, що дозволяє точно в режимі реального часу базуватись на сервісах розташування, вдосконаленому картографуванні та обробці шляхів для мешканців будинків, а також просторова аналітика для власників місць. [10]

Позиціонування в приміщенні:

- світильники, оснащені кодованим світлом або технологією Bluetooth, передають сигнали через високочастотне модульоване видиме світло (VLC) або радіохвилі на смартфон пасажира;
- додаток із підтримкою IPS розшифровує сигнали та розміщує пасажира в їх точному положенні на конкретній карті місця проведення.

Шляхи наведення:

- відправляє користувачів з їх поточного положення;
- здатний вирішувати унікальні потреби, пов'язані з вимогами доступності;
- вміщує нюанси внутрішніх маршрутів, обробку змін підлоги, сходів, ескалаторів, пандусів та ліфтів.

Місце конкретного пошуку:

- з урахуванням кожного місця та має передбачуваний автоматичний завершення пошуку та оптимізовані пріоритети пошуку;
- надає результати, класифіковані на основі близькості до місця розташування користувача або на основі частини карти, яка переглядається.

Картографування як послуга:

- гнучка та вдосконалена платформа для картографування в приміщенні, призначена для масштабування від однієї будівлі до тисяч майданчиків, кожна з індивідуальними стилями;
- підтримка розташування продуктів, послуг, місць чи будь-якого іншого типу об'єктів чи даних на карті місця проведення.

Просторовий аналіз:

- досконала платформа Location Analytics надає роздрібним торговцям інформацію про місцезнаходження, відділи, сфери обслуговування, місця розташування товарів і брендів тощо;
- параметри зйомки часу та розташування Analytics для колекційних точок даних;
- місцезнаходження, регіон та статистика підприємств;
- карта прив'язує кожен продукт, місце угоди та покупок, а також розташування всіх заходів у магазині.

### **2.2.3 IndoorAtlas**

SDK інтегрується з додатком та датчиками розумного пристрою. Кожне джерело даних - будь то магнітний, маяковий, Wi-Fi чи будь-який інший датчик - умовно дотримується. Ця інформація про середовище разом із виведеним відносним рухом користувача потім порівнюється з цифровими картами, виготовленими алгоритмами машинного навчання на хмарній платформі, для отримання точного розташування користувача. Наскільки кожне джерело даних вносить свій внесок у

цей кінцевий продукт, індивідуально розглядається на основі ймовірнісних моделей та постійно змінюваної точності джерел даних. [11]

Завдяки значущому характеру алгоритмів позиціонування видно, що чим більше даних спостерігали, тим кращі оцінки - нарешті, буде отримано бажану точність останнього метра.

Розрахунок загибних пішоходів. Технологія синтезу датчиків відстежує рух користувача за допомогою вбудованих інерційних датчиків у смартфоні. Залежно від апаратного забезпечення пристрою та бажаної точності IndoorAtlas автоматично вибирає найкращі алгоритми для пристрою та випадку використання - будь то локалізація на основі карти, налагодження способу, AR-ігри або відстеження активів. Окрім двоходового руху, охоплюються всілякі рухи від кочення до польоту.

Магнітне поле Землі. Точність останнього метра. Всі сучасні будівлі мають унікальний магнітний пейзаж, що утворюється магнітним полем Землі, який взаємодіє зі сталлю та іншими матеріалами, знайденими в конструкціях будівель. Використовуючи вбудований магнітний датчик (компас), а також інші технології зондування всередині смартфона, програмне забезпечення здатне використовувати магнітне поле всередині будівлі як карту для точного визначення та відстеження місцезнаходження людини в приміщенні, створюючи «синю точку» на карті.

Bluetooth маяки. BLE маяки стають все більш поширеними в більшості комерційних середовищ. Вони забезпечують швидкий та економічно ефективний спосіб створити невеликі розміри для розміщення в приміщенні для розміщення позицій як для iOS, так і для Android. На жаль, маяки мають слабкі місця, коли справа стосується точності та масштабованості.

Технологія IndoorAtlas використовує будь-які розгортання маяка і виправляє ці слабкі сторони, доповнюючи інформацію з інших джерел, таких як геомагнітне поле. Це зменшує необхідну кількість маяків, при цьому все ще пропонує високу точність розташування. Для IndoorAtlas, і особливо для iOS, маяки забезпечують швидку ефективність часу до першого встановлення та допомогу у виявленні переходів підлоги та приміщень. Підтримуються стандарти iBeacon та Eddystone.

Сигнали Wi-Fi. Триангуляція Wi-Fi - одна з найдавніших технологій, що застосовується при розміщенні в приміщенні. І з поважної причини. Багате середовище Wi-Fi існує у торгових центрах, офісах та житлових будинках. Також майже всі мобільні пристрої сьогодні мають чіп Wi-Fi. Ця вільно доступна інформація може забезпечити точність рівня кімнати за добрих обставин. Хоча відстеження магнітного поля забезпечує точність системи внутрішнього позиціонування в приміщенні IndoorAtlas, Wi-Fi забезпечує швидку ефективність часу до першого виправлення та допомогу у виявленні переходів підлоги та закритих приміщень. Ми підтримуємо всі мережі Wi-Fi, з якими можна спілкуватися сучасним смартфоном.

Барометричний тиск. Барометричний датчик смартфона може виявити навіть найменші зміни атмосферного тиску. Зазвичай це трапляється, коли користувач рухається у вертикальному напрямку, наприклад, піднімаючись по сходах вгору або вниз, або використовує ескалатори або ліфти. Оскільки тиск всередині будівлі змінюється з години на годину, неможливо вивести будь-яку абсолютну інформацію тільки з цього датчика. Коли барометричний датчик доступний, ми використовуємо його показання як одне джерело даних, щоб вказати, що користувач, можливо, перейшов з одного рівня на інший. Якщо ця інформація викликає подію зміни підлоги, залежить від того, решта джерел даних згодні чи ні. Хмара IndoorAtlas: найкраща з обох світів SDK IndoorAtlas спілкується з нашою хмарною платформою, яка зберігає сигнальні карти. Наш хмарний сервіс пропонує вам та вашим клієнтам найсучасніші алгоритми та статистику позиціонування.

## **Висновки до розділу 2**

Отже, проаналізувавши інформацію існуючих систем та ресурсів, що використовуються для локалізації та картографії, можемо зробити підсумки.

Існуючі системи використовують технології доповненої реальності для взаємодії з користувачем. Для точної геопозиції необхідні додаткове обладнання та більше доопрацювання алгоритму. Оцінка та висновки наведені в таблиці 2.1.

Таблиця 2.1 — Порівняння схожих систем

	Запропонована система	Inplaces	Hubbell IPS	IndoorAtlas
Використання AR	+	+	-	-
Використовувані платформи	ARCore	ARCore, ARKit	-	-
Оцінка точності локалізації	До 10 см	До 1 см	До 1 см	До 1-3 метрів. Чим більше повторення маршруту, тим більша точність
Додаткове обладнання	Використовуються окуляри доповненої реальності	Не потребує	Bluetooth	Bluetooth beacons, Wi-Fi

Можемо зробити висновок, що запропонована система має більшу актуальність, так як надає високу точність, є доступною для звичайного користувача, не використовує додаткового обладнання



## **3 ОПИС ПОБУДОВАНОЇ МОДЕЛІ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ**

В розділі описуються використані програмні засоби та обґрунтовуються обрані рішення. Також пояснюються обчислення, за якими працює алгоритм коректного відрисовування схеми будівлі на сцені.

Пропонується до розгляду ключові рішення поставленої задачі.

### **3.1 Бібліотека netDXF**

Схема будівлі створюється в додатку AutoCAD з використанням таких об'єктів як лінії, дуги, кола та інші. Для коректного відображення даних потрібно враховувати одиниці довжини ліній, що утворюють стіни, щоб схема мала фіксований розмір, по якому буде рахуватися найкоротший шлях відносно позиції користувача на карті приміщення в додатку.

Обов'язковою умовою є з'єднання ліній, адже якщо вони не будуть з'єднані – карта не буде мати потрібної цілісності и не буде коректно відрисована на сцені додатку.

Також, має бути враховано, що відрисовуватися повинні лише лінії, адже кола містять метадані, що є своєрідним маркером, до якого прив'язуються точки для побудови маршруту користувачеві.

Дуги – це прив'язка до маркера, що встановлює позицію користувача відносно карти. Було прийняте рішення використовувати для маркера саму дугу, щоб відобразити з якого боку стіни має знаходитися маркер. Тобто лицьовою стороною маркеру є саме випукла сторона дуги. Тож сама дуга містить в собі лише інформацію про те, що в даному місці знаходиться маркер та його унікальний ідентифікатор, що використовується при генерації маркеру в додатку.

Також об'єкт коло може містити не лише дані про номер аудиторії, а такожчи об'єкт не є сходами. Якщо ж об'єкт сходи – він має відповідну позначку з можливим

напрямком. Наприклад, якщо сходи знаходяться на третьому поверсі, об'єкт міститиме такі метадані: STAIRS: 2,4, що означає тип об'єкту та поверхи, на які можна перейти.

Також, необхідно враховувати, що завантажена схема у вигляді файлу має 2D-формат, тож не може містити висоту і товщину стін. Дана умова не задовольняє тип додатку, адже при використанні як режиму перегляду карти, так і режиму камери, використовується тривимірна модель відображення з використанням доповненої реальності.

Тож, при спрацюванні алгоритму відрисовування схеми на сцені додатку, товщина та висота стін відображаються у відповідності з заданими даними. Стіни з'єднуються по внутрішній видрисованій лінії об'єкту, що зменшує можливість накопичення похибок.

Бібліотека DXF забезпечує підтримку для генерації файлів DXF для використання з програмами CAD, використовуючи стандартні команди "малювання" та "заповнення" графіки Graphics. Бібліотека надає спеціальний підклас Graphics2D, DXFGraphics, який асоціюється з документом DXF і який робить команди малювання в синтаксисі DXF. Послідовність стандартних методів малювання Graphics для екземпляра DXFGraphics таким чином створить структурований документ DXF, який при відкритті зі стандартною програмою CAD буде відображати зображення як CAD-дизайн, який можна змінювати в рамках програми CAD. [12]

Клас DXFGraphics реалізує графічні операції, визначені стандартним класом Java Graphics2D. Щоб створити малюнок DXF, ви використовуєте стандартні дзвінки для малювання Java Graphics (включаючи малювання, перетворення тощо) на екземплярі DXFGraphics. DXFGraphics буде кодувати ці команди малювання Java як об'єкти DXF в асоційованому DXFDocument, які потім можна отримати у вигляді текстового рядка та зберегти у файлі DXF.

DXFGraphics асоціюється з DXFDocument.

Бібліотека DXF забезпечує більшість операцій малювання, доступних у стандартних класах Graphics і Graphics2D. Однак деякі методи в даний час є нереалізованими (деякі через відсутність підтримки в DXF). Ці нереалізовані операції

будуть ігноровані або викинуть `UnsupportedOperationException`, якщо використовується, як зазначено нижче. Деякі додаткові функції наразі не мають або обмежують підтримку, як зазначено нижче.

Підтримувані операції:

- `drawLine`, `drawPolyline`;
- `drawPolygon`, `fillPolygon`;
- `drawArc`, `fillArc`;
- `drawOval`, `fillOval`;
- `drawRect`, `fillRect`, `clearRect`;
- `drawRoundRect`, `fillRoundRect`;
- малювати (`Shape`), заливати (`Shape`), очищати (`Shape`).

Текстові операції:

- `drawString`, `drawChars`, `drawBytes`;
- `registerFont`.

Властивості малювання:

- колір;
- ширина лінії;
- властивості шрифту: `serif` / `sans-serif`, розмір, напівжирний, курсив.

Операції трансформації аффіну:

- масштаб;
- обрізання;
- переклад;
- обертання.

Бібліотека надає методи створення та малювання В-сплайнів на стандартній `Java Graphics`, а також генерування коду `DXF` для сплайнів.

Не підтримується:

- Нанесення зображення - кидає `UnsupportedOperationException`;
- Фарба `xfermodes` – проігноровано;
- Відсікання, включаючи `clipRect`, `clipRegion`, `clipPath` тощо. – Ігнорується.

Бібліотека JDxF додає елементи DXF, які не вимагаються стандартом, але необхідні для відкриття файлів у AutoCAD.

Формат DXF - це теґоване представлення даних всієї інформації, що міститься в AutoCAD файл малювання. Теґовані дані означають, що кожному елементу даних у файлі передуює ціле число, яке називається кодом групи. Значення коду групи вказує, який тип даних елемент впливає. Це значення також вказує значення елемента даних для даного об'єкта (або запис) типу. Практично вся інформація, вказана користувачем, у файлі малюнка може бути представлена у форматі DXF.

Кожен груповий код, зазначений у цій посиланні, представлений числовим кодом групи значення та опис. Усі коди груп можуть застосовуватися до файлів, програм DXF™ (AutoLISP або ObjectARX), або те й інше. Коли опис коду відрізняється для додатків та файлів DXF (або стосується лише одного чи іншого), опису передують такі показники:

- опис програми, що стосується програми;
- опис файлу DXF.DXF.

Якщо опис є загальним для файлів і програм DXF, індикатора немає надається. Необов'язкові коди вказуються як "необов'язкові" в описі.

У форматі DXF™ визначення об'єктів відрізняється від сутності: об'єкти не мають графічного подання та сутності. Наприклад, словники є об'єктами, а не сутностями. Суб'єкти називають також графічними об'єктами в той час як об'єкти відносять до неграфічних об'єктів. Суб'єкти відображаються як у розділі BLOCK, так і ENTITIES у файлі DXF. The використання групових кодів у двох розділах ідентично. Завжди відображаються деякі групові коди, які визначають сутність; інші - необов'язкові і відображаються лише в тому випадку, якщо їх значення відрізняються від значень за замовчуванням. Не пишіть програми, які покладаються на дане тут порядок. Кінець суб'єкта господарювання позначається наступною групою 0, яка починає наступну сутність або вказує на кінець розділу.

Розміщення файлів DXF з майбутніх версій AutoCAD® буде легше якщо ви пишете програму обробки DXF керованою таблицею, ігноруйте невизначені групові коди і не допускайте припущення щодо порядку групових кодів у сутність. З кожним

новим випуском AutoCAD до об'єктів додаватимуться нові коди груп для розміщення додаткових функцій.

У файлах DXF (їх можна відкрити в Блокноті) дані зберігаються у дворядкових (дволінійних, подвійних, сполучених, як би ви їх не називали) структурах. Перший рядок містить своєрідний ідентифікатор (або тег), а другий рядок містить значення даних. Отже, щоб інтерпретувати файл DXF, ми повинні інтерпретувати рядки два на два.

Залежно від цієї подвійної структури, ми можемо інтерпретувати всі об'єкти, які перебувають у файлі DXF.

Але в даний час тільки LINE, LWPOLYLINE, CIRCLE і ARC об'єкти можуть бути впізнаваними.

### **3.2 Запис метаданих з використанням XDATA AutoCAD**

Бувають випадки, коли було б непогано мати власні програми зберігати значення, а потім отримати їх пізніше. Хоча ви можете використовувати глобальну змінну, щоб тимчасово зберігати значення при цьому спеціальна програма залишається в пам'яті, глобальні змінні не зберігаються через кілька сеанси. Використовуючи бібліотеку об'єктів AutoCAD та VBA, ви можете зберігати значення таким чином, щоб вони зберігалися між сеансами малювання або AutoCAD. Якщо ви хочете, щоб значення було доступне, коли малюнок відкритий, ви можете використовувати розширені дані або користувацький словник.

XDATA може бути приєднаний до об'єкта як спосіб диференціювати один об'єкт від іншого або, у деякі випадки впливають на те, як об'єкт може виглядати в зоні малювання. Значення можуть бути збережені в Реєстрі Windows і витягнуті з будь-якого сеансу AutoCAD, який ваша спеціальна програма завантажена в. Значення, що зберігаються в Реєстрі Windows, можуть представляти рядки, 2D або 3D точки, цілі числа та парні. В якості альтернативи можна записати значення тоа текстовий файл і читати пізніше.

Розширення інформації про об'єкти Кожен об'єкт на кресленні має заздалегідь встановлений набір властивостей, які визначають, як цей об'єкт повинні з'являтися або поводитися. Наприклад, ці властивості використовуються для визначення розміру кола або розташування рядка в кресленні, хоча ви не можете використовувати VBA для додавання нового ресурсу до об'єкта, ви можете додати спеціальну інформацію до об'єкта. Спеціальна інформація, яка є доданий до об'єкта відомий як XDATA. [13]

XDATA структурується за допомогою двох масивів. Перший масив містить типи даних для значень зберігатись (коди групи DXF); другий масив містить значення, що підлягають збереженню. Два масиви повинна містити однакову кількість елементів. Як частина значень, що підлягають збереженню, перше значення має бути ім'ям програми для ідентифікації користувачької програми, яка додала XDATA. Після назва програми, масив може містити будь-які підтримувані значення. Підтримувані значення - це рядки, цілі числа, подвійні і назви юридичних осіб, серед інших. Значення, що складають XDATA, і те, що вони представляють, залежить від вас, творця дані. Дані в масивах XDATA можуть бути використані для визначення місця, де об'єкт слід розмістити чи який шару, на якому він повинен знаходитися, для зберігання інформації про зовнішній запис бази даних, який пов'язаний з об'єктом або будувати зв'язки між об'єктами на кресленні.

Спосіб використання або використання даних залежить від програміста. Крім Xdata, графічні та неграфічні об'єкти підтримують те, що називають розширенням словників, словники розширень – це на зразок таблиць записів, які можна приєднати до об'єкта. Наприклад, ви можете зберігати історію редагування малюнка в словнику розширень, який приєднати до простору моделі, а потім заповнити блок заголовка креслення цією інформацією. Навіть AutoCAD використовує словники розширень для реалізації функцій, таких як стани шару та фільтри, які додаються до таблиці символів шару.

Робота з XDATA Додавання XDATA до об'єкта вимагає зробити початкове планування та виконати кілька кроків. У наступному списку викладено кроки, які необхідно виконати для прикріплення XDATA до об'єкта:

- потрібно перевірити, чи до об'єкта вже додано XDATA та яке ім'я програми. Якщо Xdata вже додано до назви програми, яку ви планували використовувати, перейдіть до «Заміна»;
- визначити та зареєструвати назву програми для користувальницької програми;
- визначити масив, який буде містити коди групи DXF, які визначатимуть типи даних масив значень даних. Першим елементом у масиві має бути 1901, що являє собою код групи DXF для імені програми;
- визначити масив, який буде містити значення даних для Xdata. Перший елемент масиву повинен бути рядок, що представляє ім'я програми;
- отримати об'єкт, до якого ви хочете додати XDATA;
- додати XDATA до об'єкта методом SetXData.

Перш ніж додавати XDATA, слід перевірити, чи об'єкт уже має XDATA з ім'ям програми, що додається до нього. Якщо це так, слід замінити XDATA з новим.

Потрібно виконати наступні кроки, щоб змінити XDATA, раніше прикріплений до об'єкта:

- визначити значення, які становлять XDATA;
- визначити масив, який буде містити коди групи DXF, які будуть використовуватися для представлення типи даних масиву значень даних; перший елемент масиву повинен бути 1001, що представляє код групи DXF для імені програми;
- визначити масив, який буде містити значення даних для XDATA; перший елемент масиву повинна бути рядок, що представляє ім'я програми;
- отримати об'єкт, на який потрібно замінити XDATA.

Масив, який визначає типи даних значень у XDATA, повинен визначатися як integer data type, тоді як масив значень даних для XDATA повинен визначатися як variant data type. Після того, як масиви, що становлять XDATA, будуть визначені, XDATA можна приєднати до об'єкта методом SetxData.

Далі показано синтаксис методу SetXData:

`object.SetXData data type, dataValues`

Його аргументи такі:

- `object` – аргумент об'єкта представляє об'єкт AutoCAD, який ви хочете долучити;
- `Data type` – аргумент `dataTypes` – це масив цілих чисел, що представляють типи даних значення, що зберігаються з XDATA об'єкта;
- `dataValues` – аргумент `dataValues` – це масив варіантів, що представляють значення даних зберігатись з XDATA об'єкта.

Після того, як XDATA буде прикріплена до об'єкта, може знадобитися оновлення об'єкта, якщо XDATA впливає на зовнішній вигляд об'єкта.

### 3.3 Маркери ArUCO

Для коректної роботи підсистеми позиціонування, необхідно, щоб схема містила записані маркери для оцінки положення відносно схеми. Так як позиціонування за допомогою додаткового обладнання, такого як додаткові маяки чи датчики, використання NFC-технології і є доволі складним у впровадженні в додатку, було вирішено використати технологію маркерів ArUCO. Це безкоштовна технологія, що потребує підключення бібліотеки `aguco` та додаткових налаштувань в коді. [14]

Спершу додається маркер за допомогою метаданих, записується в об'єкт на схемі з відповідним унікальним ідентифікатором, після чого метадані оброблюються підсистемою інтеграції тривимірної побудови карти будівлі і відповідно відрисовується на сцені додатку з видимим маркером. Під час обробки метаданих генерується маркер ArUCO, що необхідно роздрукувати, він є унікальним, тож переміщенні маркеру фізично вплине на коректність позиціонування в додатку, дані можуть бути недостовірними. Розглянемо більш докладно обрану технологію та процес обробки і отримання даних для позиціонування користувача на карті в додатку.

Оцінка пози має велике значення у багатьох додатках для комп'ютерного зору: навігації роботів, доповненої реальності та багатьох інших. Цей процес заснований на знаходженні відповідностей між точками в реальному середовищі та їх проекції 2d



зображення. Зазвичай це важкий крок, і тому звичайне використання синтетичних або довідкових маркерів для спрощення. Один з найпопулярніших підходів - використання двійкових квадратних фідуціальних маркерів. Основна перевага цих маркерів полягає в тому, що один маркер забезпечує достатню відповідність (чотири його кути), щоб отримати позу камери. Також внутрішня двійкова кодифікація робить їх особливо надійними, що дозволяє можливість застосовувати методи виявлення та виправлення помилок. Модуль aruco базується на бібліотеці ArUco , популярній бібліотеці для виявлення квадратних фідуціальних маркерів, розробленій Рафаелем Муньосом та Серхіо Гаррідо.

Маркер ArUco - це синтетичний квадратний маркер, складений широкою чорною рамкою та внутрішньою бінарною матрицею, що визначає його ідентифікатор (id). Чорна облямівка полегшує її швидке виявлення в зображенні, а двійкова кодифікація дозволяє визначити і застосувати методи виявлення та виправлення помилок. Розмір маркера визначає розмір внутрішньої матриці. Наприклад, розмір маркера 4x4 складається з 16 біт (рисунок 3.1).

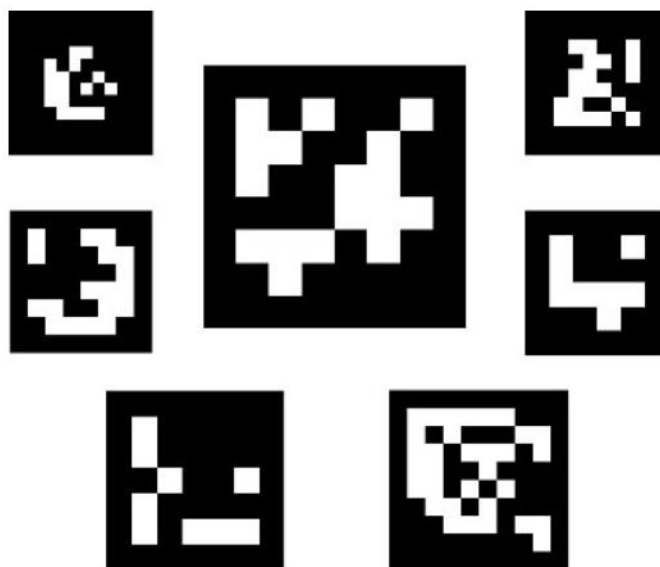


Рисунок 3.1 – Приклади маркерів

Слід зазначити, що маркер можна знайти повернутим у навколишньому середовищі, проте процес виявлення повинен бути в змозі визначити його початкове

обертання, щоб кожен кут був визначений однозначно. Це також робиться на основі двійкової кодифікації. Словник маркерів - це набір маркерів, які розглядаються в конкретній програмі. Це просто перелік двійкових кодифікацій кожного з його маркерів. Основними властивостями словника є розмір словника та розмір маркера. Розмір словника - це кількість маркерів, що склали словник. Розмір маркера - це розмір цих маркерів (кількість біт). Модуль aruco включає деякі заздалегідь задані словники, що охоплюють діапазон різних розмірів словника та розмірів маркерів. Можна подумати, що ідентифікатор маркера - це число, отримане при перетворенні двійкової кодифікації в десяткове базове число. Однак це неможливо, оскільки для високих розмірів маркерів кількість бітів занадто велика, а керування такою великою кількістю не є практичною. Натомість ідентифікатор маркера - це просто індекс маркера всередині словника, до якого він належить. Наприклад, перші 5 маркерів всередині словника мають ідентифікатори: 0, 1, 2, 3 і 4. [15]

Перед їх виявленням маркери потрібно роздрукувати, щоб розмістити їх у навколишньому середовищі. Зображення маркерів можна генерувати за допомогою drawMarker() функції. Наприклад, давайте проаналізуємо такий виклик:

```
cv::Mat      markerImage;      cv::aruco::Dictionary      dictionary      =  
cv::aruco::getPredefinedDictionary(cv::aruco::DICT_6X6_250);  
cv::aruco::drawMarker(dictionary, 23, 200, markerImage, 1);
```

По-перше, Dictionary-об'єкт створюється шляхом вибору одного із заздалегідь заданих словників модуля aruco. Конкретно, цей словник складається з 250 маркерів і розміром маркера 6x6 біт ( DICT\_6X6\_250).

Параметри drawMarker:

- перший параметр - це Dictionary об'єкт, створений раніше;
- другий параметр - ідентифікатор маркера, в цьому випадку маркер 23 словника DICT\_6X6\_250. Кожен словник складається з різної кількості маркерів. У цьому випадку дійсні ідентифікатори мають значення від 0 до 249. Будь-який конкретний ідентифікатор із допустимого діапазону призведе до виключення;
- третій параметр, 200, - це розмір зображення вихідного маркера. У цьому випадку вихідне зображення матиме розмір 200x200 пікселів. Цей параметр

повинен бути досить великим, щоб зберігати кількість бітів для конкретного словника. Так, наприклад, ви не можете генерувати зображення розміром 5x5 пікселів для розміру маркера 6x6 біт (і це не враховуючи межу маркера). Крім того, щоб уникнути деформацій, цей параметр повинен бути пропорційним кількості біт + розмір межі або, принаймні, значно перевищувати розмір маркера (як 200 у прикладі), так що деформації будуть незначними;

- четвертий параметр - вихідне зображення;
- нарешті, останній параметр - необов'язковий параметр для визначення ширини чорної межі маркера. Розмір задається пропорційно кількості бітів. Наприклад, значення 2 означає, що межа буде мати ширину, еквівалентну розміру двох внутрішніх біт. Значення за замовчуванням - 1.

З огляду на зображення, де видно деякі маркери ArUco, процес виявлення повинен повернути список виявлених маркерів. Кожен виявлений маркер включає: Розташування чотирьох кутів на зображенні (у їх первісному порядку).

Процес виявлення маркера складається з двох основних етапів: Виявлення кандидатів у маркери (рисунок 3.2).

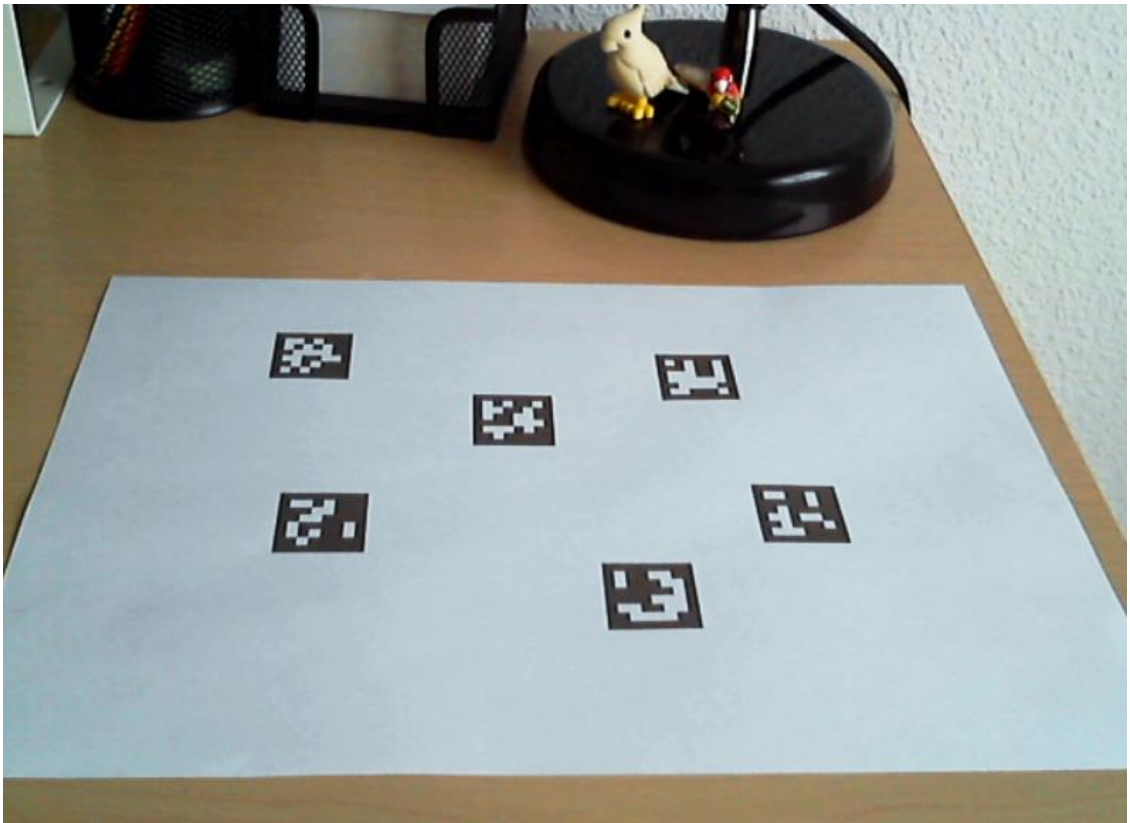


Рисунок 3.2 – Оригінальний вид маркерів до розпізнавання

На цьому етапі зображення аналізується, щоб знайти квадратні форми, які можуть бути маркерами. Він починається з адаптивного порогування для сегментації маркерів, потім контури витягуються із зображення порогового значення, а ті, що не опуклі або не наближені до квадратної форми, відкидаються. Також застосовується додаткова фільтрація (видалення занадто малих або занадто великих контурів, видалення контурів занадто близько один до одного тощо).

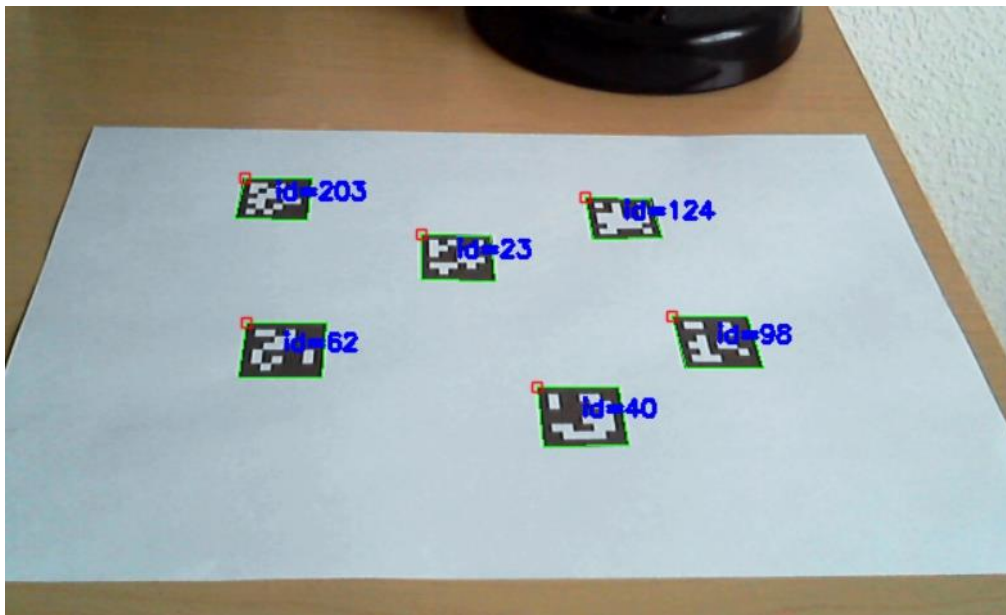


Рисунок 3.3 – Виявлення маркера

Після виявлення кандидата необхідно визначити, чи є вони насправді маркерами, проаналізувавши їх внутрішню кодифікацію (рисунок 3.3). Цей крок починається з вилучення бітів маркера кожного маркера. Для цього спочатку застосовується перспективне перетворення для отримання маркера в його канонічній формі. Потім канонічне зображення порогове значення, використовуючи Otsu для розділення білих і чорних бітів. Зображення поділяється на різні комірки відповідно до розміру маркера та розміру рамки та кількості чорних чи білих пікселів на кожній комірці, що підраховується, щоб визначити, чи є білим чи чорним шматочком. Нарешті, біти аналізуються, щоб визначити, чи належить маркер конкретному словнику, і при необхідності застосовуються методи виправлення помилок. [16]

Для оцінки пози на камері вам потрібно знати параметри калібрування камери. Це матриця камери та коефіцієнти спотворень. Також можна калібрувати камеру, використовуючи модуль агусо, як це пояснено в навчальному посібнику з калібрування. Це потрібно зробити лише один раз, якщо не буде змінена оптика камери (наприклад, зміна фокусу). Зрештою, те, що отримується після калібрування, матриця камери: матриця 3x3 елементів із фокусними відстанями та центральними координатами камери (також внутрішніми параметрами) та коефіцієнтами спотворень: вектор з 5 елементів або більше, що моделює модель спотворення,

спричинене камерою. Оцінюючи позу маркерами  $ArUco$ , можна оцінювати позу кожного маркера окремо. Якщо потрібно оцінити одну позу з набору маркерів, потрібно скористатися дошками  $agiso$ . Пози камери щодо маркера - це 3d-трансформація від системи координат маркера до системи координат камери. Він задається вектором обертання та перекладом.

### **3.4 Реалізація повороту карти на сцені додатку**

Окрім того, що карта має бути відрисована у відповідності до 2D схеми та перенесені всі метадані, карта, відрисована в додатку повинна змінюватися при переміщенні користувача, чи при зміні точки зору додатку.

При чому постає питання як саме реалізувати поворот карти зі збереженням розміщення побудованих об'єктів відносно один одного.

#### **3.4.1 Визначення орієнтації**

Термін "орієнтація" має на увазі, що ми знаходимося в деякій заданій системі відліку. Наприклад, фраза "він повернув голову вліво" осмислена тільки тоді, коли ми представляємо, де знаходиться "ліво" і де перебувала до цього голова. Це важливий для розуміння момент, адже якби це був монстр з головою на животі верхівкою вниз то фраза "він повернув голову вліво" вже не здається такою однозначною.

Трансформацію, яка певним чином обертає з однієї орієнтації в іншу, назовемо поворотом. За допомогою повороту можна описати і орієнтацію об'єкта, якщо ввести якусь орієнтацію за замовчуванням як точку відліку. Наприклад, будь-який об'єкт, описаний за допомогою набору трикутників, вже має орієнтацію за замовчуванням.

Координати його вершин описуються в локальній системі координат цього об'єкта. Довільну орієнтацію цього об'єкта можна описати матрицею повороту щодо його локальної системи координат. Також можна виділити таке поняття як "обертання". Під обертанням будемо розуміти зміну орієнтації об'єкта заданим чином

в часі. Щоб однозначно задати обертання, треба, щоб в будь-який момент часу ми могли визначити точну орієнтацію обертається об'єкта. Іншими словами обертання задає "шлях", пройдений об'єктом при зміні орієнтації. У такій термінології поворот не задає однозначного обертання об'єкта. Важливо розуміти що, наприклад, матриця не задає однозначного обертання тіла, одну і ту ж матрицю повороту можна отримати, повернувши об'єкт на 180 градусів навколо фіксованої осі і на  $180 + 360$  або  $180 - 360$ . Ці терміни я застосовую для демонстрації відмінностей в поняттях, і ні в якій мірі не наполягаю на використанні. Надалі залишу за собою право говорити "матриці обертання". [17]

При слові орієнтація часто виникають асоціація з напрямком. Часто можна почути фрази подібні "він повернув голову в бік наближається локомотива". Наприклад, орієнтацію автомобіля можна було б описати напрямком, в якому дивляться його фари. Однак напрямок задається двома параметрами (наприклад, як в сферичній системі координат), а об'єкти в тривимірному просторі мають три ступені свободи (обертання). У випадку з автомобілем він може дивитися в одному напрямку як стоячи на колесах, так і лежачи на боці або на даху. Орієнтацію дійсно можна задати напрямком, але їх потрібно два. Давайте розглянемо орієнтацію на простому прикладі голови людини.

Домовимося про вихідне положення, в якому голова орієнтована за замовчуванням (без обертання). За вихідне приймемо положення, в якому голова дивиться особою у напрямку осі "z", а вгору (верхівкою) дивиться в напрямі осі "y". Назвемо напрямок, в якому повернуто обличчя "dir" (без обертання збігається з "z"), а напрямок, куди дивиться верхівка "up" (без обертання збігається з "y"). Тепер у нас є точка відліку, є локальна координатна система голови "dir", "up" і глобальна з осями x, y, z. Довільно повернемо голову і відзначимо, куди дивиться обличчя. Дивлячись в цьому ж напрямку можна обертати голову навколо осі, що збігається з напрямком погляду "dir". Наприклад, нахиливши голову на бік (притулившись щокою до плеча) ми будемо дивитися в тому ж напрямку, але орієнтація голови зміниться. Щоб зафіксувати поворот навколо напрямку погляду, використовуємо ще й напрямок "up"

(направлено до верхівки). В цьому випадку ми однозначно описали орієнтацію голови і не зможемо її повернути, не змінивши напрямки осей "dir" і "up" (рисунок 3.1). [18]

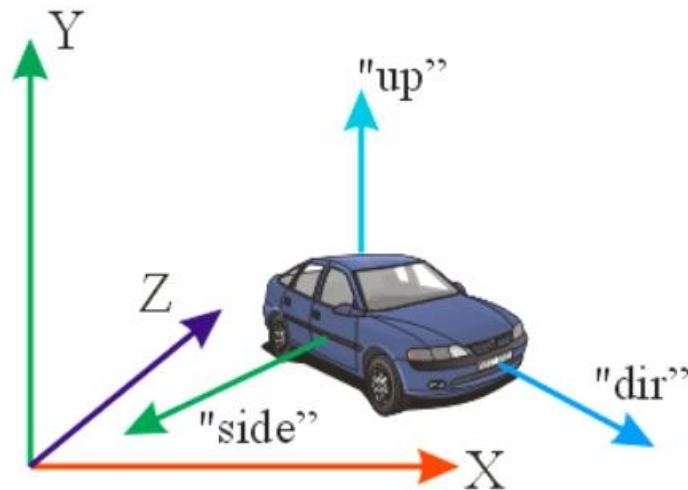


Рисунок 3.4 – Осі "dir" і "up" при повороті відносно осі координат

Ми розглянули досить природний і простий спосіб завдання орієнтації за допомогою двох напрямків. Як же описати наші напрямки в програмі, щоб ними було зручно користуватися? Простий і звичний спосіб зберігати ці напрямки у вигляді векторів. Опишемо напрямки за допомогою векторів довжиною в одиницю (одиничних векторів) в нашій глобальній системі координат хуз. Перший важливе питання, як би наші напрямки передати в зрозумілому вигляді графічного API? Графічні API працюють в основному з матрицями. Нам би хотілося отримати матрицю повороту з наявних векторів. Два вектора описують напрямки "dir" і "up" і є та сама матриця повороту, а точніше два компоненти матриці повороту  $3 \times 3$ . Третій компонент матриці ми можемо отримати з векторного добутку векторів "dir" і "up" (назвемо його "side"). У прикладі з головою вектор "side" буде дивитися в напрямку одного з вух. Матриця повороту це і є координати трьох векторів "dir", "up" і "side" після повороту. Ці вектори до повороту збігалися з осями глобальної системи координат хуз. Саме у вигляді матриці повороту дуже часто і зберігають орієнтацію об'єктів (іноді матрицю зберігають у вигляді трьох векторів). Матрицею можна задати орієнтацію (якщо відома орієнтація за замовчуванням) і поворот.



Схожий спосіб представлення орієнтації, називається кути Ейлера (Euler Angles), з тією лише відмінністю, що напрямок "dir" задається в сферичних координатах, а "up" описується одним кутом повороту навколо "dir". В результаті отримаємо три кути обертання навколо взаємно перпендикулярних осей. У аеродинаміці їх називають Крен, тангажу, нишпорення (Roll, Pitch, Yaw або Bank, Heading, Attitude). Крен (Roll) - це нахил голови вправо або вліво (до плечей), поворот навколо осі проходить через ніс і потилицю. Тангажу (Pitch) - це нахил голови вгору і вниз, навколо осі проходить через вуха. І нишпорення (Yaw) - це повороти голови навколо шиї. Треба пам'ятати, що повороти в тривимірному просторі не комутативні, а значить, на результат впливає порядок поворотів. Якщо ми повернемо на R1 а потім на R2, орієнтація об'єкта не обов'язково збігатиметься з орієнтацією при повороті на R2 і потім на R1. Саме тому при використанні Углов Ейлера важливий порядок поворотів навколо осей. Зверніть увагу, що математика кутів Ейлера залежить від обраних осей (ми використовували тільки один з можливих варіантів), від порядку повороту навколо них, а також від того в якій системі координат відбуваються повороти, в світовій або локальній об'єкта. У кутах Ейлера можна зберігати і обертання і поворот. [19]

Величезний недолік такого уявлення, відсутність операції комбінації повороту. Не намагайтеся складати покомпонентно кути Ейлера. Підсумковий поворот не буде комбінацією вихідних поворотів. Це одна з найпоширеніших помилок початківців розробників. Щоб повернути об'єкт, зберігаючи обертання в кутах Ейлера, нам доведеться перевести обертання в іншу форму, наприклад в матрицю. Потім перемножити матриці двох поворотів і з підсумкової матриці витягти кути Ейлера. Проблема ускладнюється ще й тим що в окремих випадках пряме додавання кутів Ейлера працює. У разі комбінації обертань навколо однієї і тієї ж осі, цей метод математично вірний. Повернувши на 30 градусів навколо осі X, а потім ще раз навколо X на 40 градусів ми отримаємо поворот навколо X на 70 градусів. У разі обертань по двох осях просте додавання кутів може давати якийсь "очікуваний" результат. Але як тільки з'являється поворот по третій осі, орієнтація починає поводитися непередбачувано. Багато розробники витрачають місяці праці щоб

змусити працювати камеру "правильно". Потрібно звернути пильну увагу до цього недоліку, особливо якщо прийняте рішення використовувати кути Ейлера для подання обертань. Початківцям програмістам здається що, використовувати кути Ейлера найпростіше. Дозволю собі висловити особисту думку, що математика кутів Ейлера набагато складніше і підступніше ніж математика кватерніонів.

Кути Ейлера це комбінація (композиція) обертань навколо базових осей. Існує ще один, простий, спосіб завдання обертання. Цей спосіб можна назвати "суміш" обертань навколо базових координатних осей, або просто обертання навколо довільної фіксованої осі. Три компоненти описують обертання утворюють вектор, що лежить на осі, навколо якої і повертається об'єкт. Зазвичай зберігають вісь обертання в вигляді одиничного вектора і кут повороту навколо цієї осі в радіанах або градусах (Axis Angle). Вибравши відповідну вісь і кут можна задати будь-яку орієнтацію об'єкта. У деяких випадках зручно зберігати кут обертання і вісь в одному векторі. Напрямок вектора в цьому випадку збігається з напрямком осі обертання, а довжина його дорівнює куту повороту. У фізиці, таким чином, зберігають кутову швидкість. Вектор, що співпадає напрямком з віссю обертання і довжиною представляє швидкість в радіанах в секунду.

### **3.4.2 Поняття кватерніону**

Кватерніони були введені в математику Вільямом Роуеном Гамільтоном (William Rowan Hamilton). Вони є хорошим інструментом для вирішення багатьох завдань, пов'язаних з тривимірним простором, і враховують його особливості, які не спостерігаються в довільних  $n$ -мірних просторах. Історія про те, як Гамільтон винайшов кватерніони, наведена тут за матеріалами книги. У 1835 році, у віці 30 років, Гамільтон навчився працювати з комплексними числами як з парами дійсних. Натхненний зв'язком між комплексними числами і двовимірної геометрією, він протягом багатьох років намагався винайти схожу алгебру, яка грала б аналогічну роль в тривимірній геометрії. [20] Те, що він шукав, на сучасному мовою називалося б тривимірної нормованої алгеброю з діленням. Причина його ускладнень полягала в

тому, що тривимірної нормованої алгебри з поділом не існує. Але його багаторічні муки все ж були винагороджені. 16 жовтня 1843 року прогулюючись з дружиною уздовж Королівського каналу по дорозі на засідання Королівської ірландської академії в Дубліні, він зробив епохальне відкриття. пізніше він згадував: «Можна сказати, я тут і зараз відчув, як електричний ланцюг думки замкнулася, а заблищали іскри виявилися фундаментальними співвідношеннями:

$$i^2 = j^2 = k^2 = i \times j \times k = -1, \quad (3.1)$$

І представленими саме в тому вигляді, в якому я їх з тих пір використовував ». Тоді ж Гамільтон зробив знаменитий акт математичного вандалізму - він вирізав ці співвідношення на кам'яних перилах мосту Брум, що ілюструється наступним малюнком (рисунок 3.1).



Рисунок 3.5 – Пам'ятна табличка на мосту Брум Бридж в Дубліні

Одна з причин, по якій ця історія настільки добре відома, полягає в тому, що з цього моменту і до кінця свого життя Гамільтон був одержимий ідеєю дослідження кватерніонів і застосування їх в геометрії. Протягом деякого часу кватерніони дійсно були в моді. У Дубліні вони в обов'язковому порядку входили в програму іспитів, а в деяких американських університетах кватерніони були єдиним изучавшимся розділом вищої математики. Багато що з того, що ми зараз робимо з скалярами і векторами тривимірного простору, робилося тоді за допомогою речових і уявних кватерніонів. Виникла ціла школа «Кватерніонщиків», яку після смерті Гамільтона

очолювали Пітер Тейт з Единбурга і Бенджамін Пірс з Гарварда. Тейт написав про кватерніонів вісім книг, в яких особлива увага приділялася додатків до фізики. Коли Гіббс винайшов сучасні позначення для скалярного і векторного добутків, Тейт обізвав їх Леонард Ейлер «виродками-гермафродитами».

Розгорнулася гаряча полеміка, в ході якої такі знаменитості, як Кельвін і Хевісайд, вибухали убивчими висловами на адресу кватерніонів. В Врешті-решт, кватерніони були переможені і придбали кілька погану репутацію, від якої вони в повній мірі так і не позбулися. Додамо, що хоча Гамільтон дійсно був першим, хто побудував кватерніони як алгебру, у кватерніонів є і більш рання історія, що починається з відкриття Ейлером в 1748 році тотожності чотирьох квадратів [21]. Крім того, видатний французький математик Б. О. Родриг в своїх дослідженнях параметризував загальний поворот за допомогою чотирьох чисел, які є фактично компонентами відповідного кватерніона. Це дозволяє розглядати його як провісника ідей Гамільтона, оскільки приводиться їм правило множення збігається з формулою Гамільтона для похідних двох кватерніонів.

### **3.4.3 Реалізація схеми та перетворення в тривимірну карту**

Зазвичай, всі схеми будівель мають вигляд лише на папері для архітектурних планів та схем пожежної безпеки з заданим масштабом. (Рис. 5.1) і їх аж ніяк не можна використовувати для навігації в додатку.



Рисунок 3.6 – типова схема будівлі в масштабі

Звичайно ж подібну схему потрібно перенести в електронний вигляд для подальшого використання.

Схема має складатися з простих об'єктів для коректного розпізнавання системою завдяки розробленому алгоритму та містити метадані для обробки даних іншими модулями системи, таких як встановлення маркеру, точки маршруту та пошуку найкоротшого шляху. (Рис. 3.8)

Розглянемо більш докладно процес створення та підготовки схеми з внесенням метаданих та ідентифікаторів маркерів, що буде використано для встановлення позиції користувача на карті та вибору кінцевої точки маршруту, а також процесом інтеграції тривимірного об'єкту на сцену додатку з використанням описаних в розділі технологій та алгоритмів.

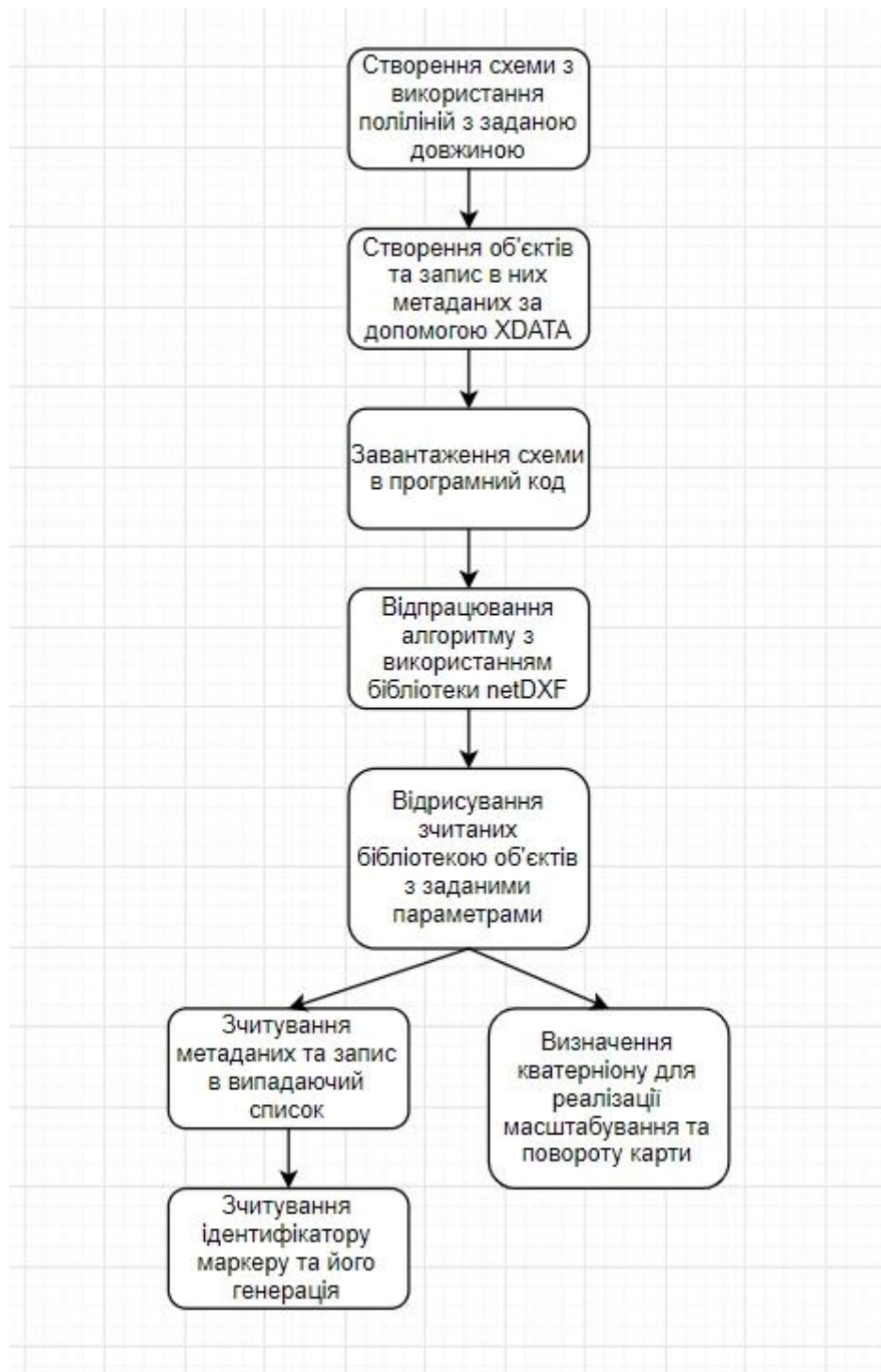


Рисунок 3.7 – Алгоритм створення та інтеграції схеми

На першому етапі створюється проста схема за допомогою поліліній, задається довжина у відповідності 1 точка = 10 см. Після цього додаються на схему об'єкти кола, за допомогою команди XDATA записуються метадані, що потім будуть оброблятися для вибору кінцевих точок призначення, що прив'язуться до об'єкту, в якому вони записані. В кола записуються два типи метаданих: номери кабінетів та

аудиторій (наприклад ROOM: 314), а також номери сходів з позначенням поверхів, на які вони ведуть (наприклад, сходи, що знаходяться на третьому поверсі STAIRS: 2,4).

При проведенні шляху, якщо необхідний пункт призначення знаходиться на іншому поверсі, користувача приведе до найближчих сходів з інструкцією переходу, після чого користувачу потрібно буде додатково відсканувати маркер біля сходів після підйому та продовжити шлях.

Маркери також відмічаються на схемі у вигляді дуги. Напрямок дуги має велике значення, адже саме за її напрямком буде відрисовано маркер в додатку (мається на увазі з якого боку стіни, внутрішнього чи зовнішнього). В об'єкті за допомогою XDATA записується ідентифікатор маркера, що буде використаний при генерації унікального ArUCO-маркеру (наприклад, MARKER:1).

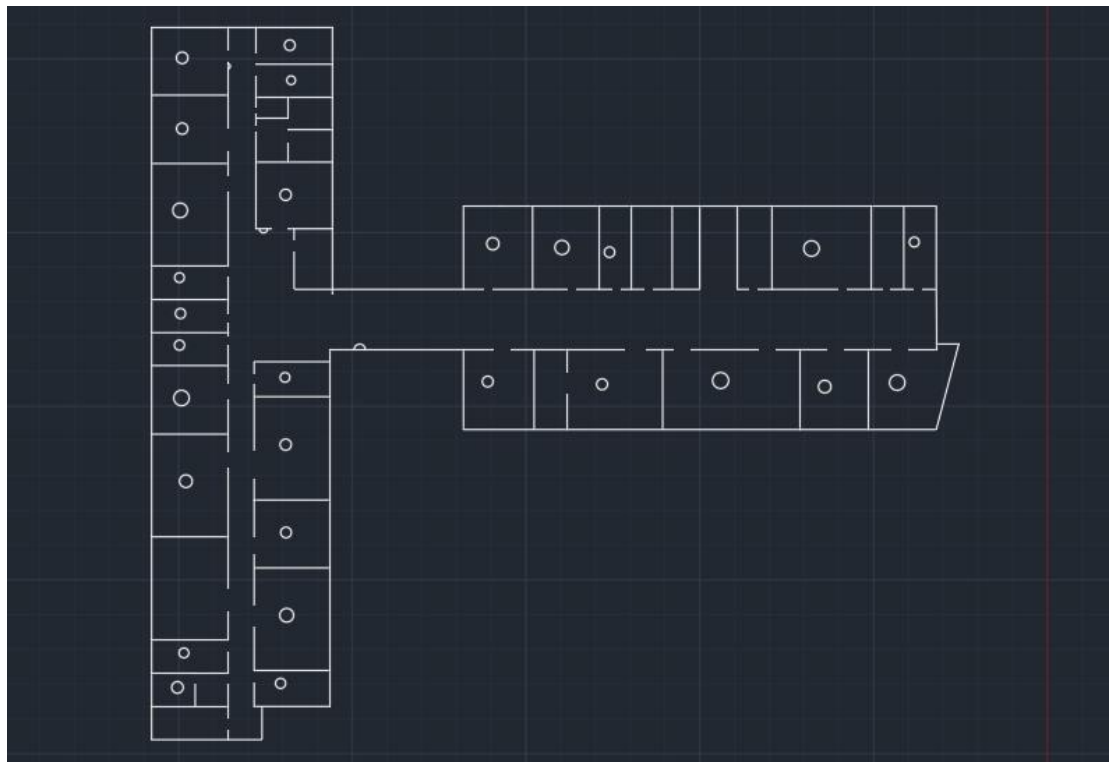


Рисунок 3.7 – Перенесена двовимірна модель схеми з використанням простих об'єктів



Після цього необхідно зберегти схему в форматі \*.dxf та завантажити в програмний код. При компіляції додатку, дані будуть регенеровані для подальшого використання за допомогою розробленого алгоритму.

Спершу алгоритм за допомогою бібліотеки netDXF зчитує прості об'єкти, в даному випадку полілінії, відрисовує їх на сцені додатку з заданою довжиною, висотою та товщиною. Товщина, довжина та висота задаються в програмному коді і є сталими значеннями. Але постає проблема – стіни будуть відрисовуватися з похибкою, що дорівнює товщині стіни. Тож потрібно враховувати, що зовнішній бік стіни має з'єднуватися з конточною іншої зовнішньої стіни. Тоді стіни будуть з'єднані коректно, для точності роботи додатку достатньо, щоб заданій на фізичній схемі довжини стін відповідали зовнішнім відрисованим стінам і довжина внутрішніх ігнорувалась, адже завдання додатку довести користувача від точки А до точки В, наприклад, до дверей конкретної аудиторії. (рисунок 3.8)

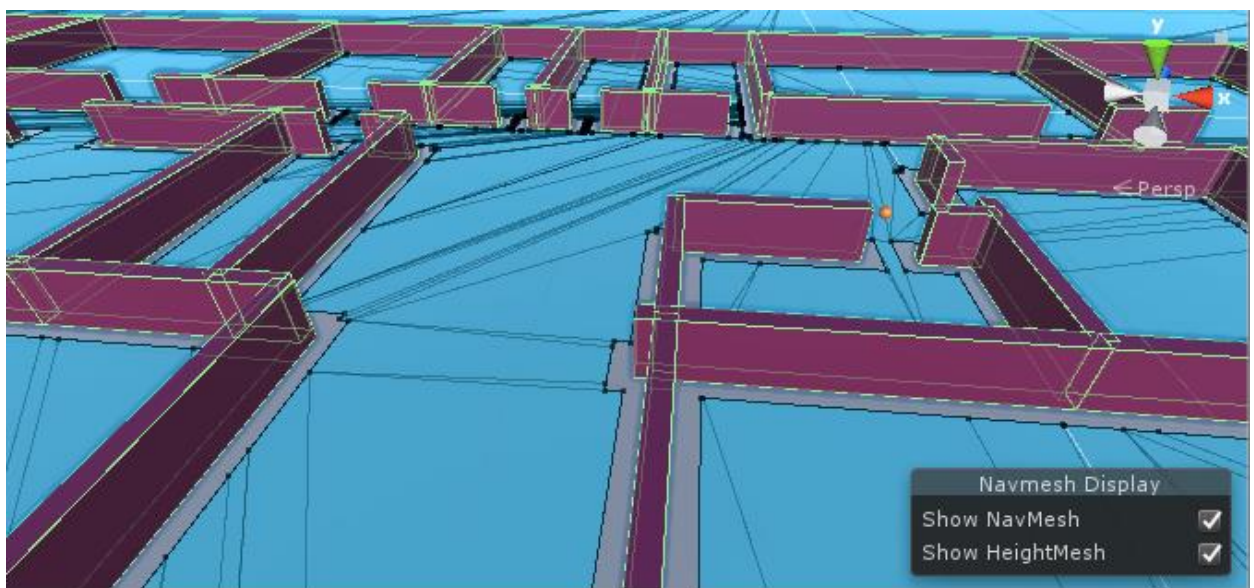


Рисунок 3.8 – Відрисована тривимірна модель

Після зчитування бібліотекою об'єктів та відрисування поліліній, алгоритм зчитує інші об'єкти, але не відрисовує їх на карті, залишаючи їх точками прив'язки. Кола – це точки прив'язки пунктів призначення, номери аудиторій завантажуються в



випадаючий список, після чого користувач має можливість обрати точку призначення (рисунок 3.9).

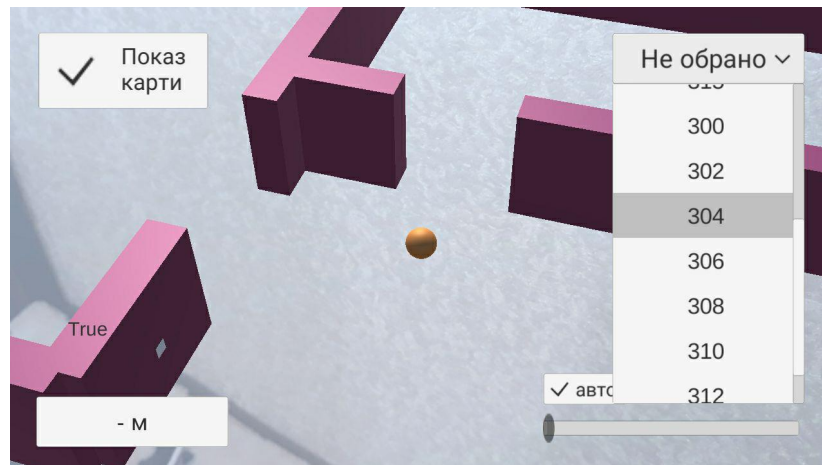


Рисунок 3.9 – Мітки в випадаючому списку, зчитані з метаданих

При перенесенні схеми в тривимірну модель, алгоритм не відрисовує кола, лише полілінії, а сприймає їх як маркери. Як показано на рисунку 3.8, метадані переносяться і записуються як мітки, що можна брати для встановлення оптимального маршруту.

Також зчитуються метадані для маркеру, генерується унікальний ArUCO-маркер по ідентифікатору та у відповідності до місця знаходження дуги та її напрямку, маркер відображається на відрисованій стіні в додатку. В тому ж місці, відповідно до схеми та додатку повинен бути розміщений фізичний маркер для зчитування його користувачем.

Після відрисування карти відбувається визначення кватерніону – це необхідно для зміни масштабу карти та куту перегляду. Без цього при масштабування карти буде змінюватися кожен об'єкт окремо, а не цілісна карта.

### Висновки до розділу 3

В розділі були детально розглянуті методи та технології реалізації підсистеми. Описані нюанси використання бібліотек для інтеграції схеми.

Для запису та зчитування схеми в додаток була використана бібліотека netDXF, що надає можливість зчитувати об'єкти з необхідною розмірністю, відрізняти їх для відрисовування на сцені додатку чи ігнорування. Також бібліотека надає можливість зчитувати метадані для використання їх в роботі з користувачем та засобів генерації маркерів.

Були розроблені засоби зчитування метаданих та використання їх для генерації маркерів. Для встановлення міток початкового шляху було прийняте рішення використовувати маркери ArUCO, так як технологія є безкоштовною та доступною.

Було розглянуто та реалізовано визначення орієнтації користувача на карті відносно маркера, а також реалізовано поворот карти при пересуванні користувача за допомогою логіки кватерніонів. Наведено основну логіку кватерніонів та переваги над кутами Ейлера. Логіка кватерніонів також надає можливість масштабувати карту в додатку та повертати самостійно під зручним кутом. Реалізація визначення кватерніону дозволила відобразити переміщення користувача по карті в реальному часі.

Запропонований додаток є простим в використанні та не потребує специфічного налаштування, чим вирізняється серед конкурентів. Все, що потрібно – це попередньо згенеровані маркери, та смартфон з камерою.

# 1 ЗАСОБИ РОЗРОБКИ

Очевидно, що для розробки такого продукту, як мобільний додаток для навігації, потрібно мати встановленим на комп'ютері потужне середовище, що має зручний редактор, дозволяє легко оперувати об'єктами та задавати їх поведінку. Саме тому для даної роботи було обрано Unity — всесвітньо відомий і один з найпопулярніших ігрових рушіїв, що дозволяє також розробляти проекти для багатьох платформ.

Крім того, ще потрібен інструментарій, який дозволяє побудувати схему будівлі, що можна завантажити в додаток для відрисовування стін будівлі на схемі, що будуть взаємодіяти з алгоритмом SLAM.

Звісно, потрібна і мова програмування для того, аби писати скрипти, які будуть визначати поведінку об'єктів додатку. Разом з Unity найчастіше використовується C#. Для написання та редагування коду, написаного цією мовою, було використано середовище розробки Microsoft Visual Studio 2017. [22]

## 1.1 Ігровий рушій Unity

Ігрові рушії, такі як Unity, — це інструменти, що стоять за іграми. Unity — це один із найпоширеніших та найпопулярніших пакетів для розробки ігор, який використовується всіма, від любителів до великих студій, для створення ігор та інтерактивних додатків для Інтернету, настільних ПК, мобільних телефонів та консолей. [23]

Unity3d є сучасним кросплатформним рушієм для створення ігор і додатків, що розроблений компанією Unity Technologies. За допомогою даного рушія можна розробляти не тільки додатки для комп'ютерів, але і для мобільних пристроїв (наприклад, на базі Android), ігрових приставок і інших девайсів.

Слід відмітити деякі важливі характеристики Unity. По-перше, в середовищі розробки Unity інтегрований ігровий рушій, що надає можливість протестувати свою гру, не виходячи з редактора. По-друге, Unity підтримує імпорт величезної кількості

різних форматів, що дозволяє розробнику гри конструювати самі моделі в більш зручному додатку, а Unity використовувати за прямим призначенням — розробки продукту. По-третє, написання сценаріїв (скриптів) здійснюється на найбільш популярних мовах програмування - C # і JavaScript. Таким чином, Unity3d є актуальною платформою, за допомогою якої можна створювати свої власні додатки і експортувати їх на різні пристрої. [24]

Для того щоб створити свою гру, вам, як мінімум, потрібно володіти одним з доступних (на Unity) мов програмування: C #, JavaScript або Boo.

Юніті — безкоштовний рушій. Обмеження — при запуску гри показується логотип Unity. З придбанням розширеної версії його можна вимкнути. Безкоштовність рушія — це те, що привернуло багатьох до розробки ігор з його використанням.

Unity — це рушій, який слід використовувати для проектів середнього розміру. Для маленьких ігор чи додатків використання такого серйозного рушія не виправдане. З іншого боку, в надто великих проектах краще застосувати інший рушій, адже Unity не надто швидкий через інтерпретовану мову C# та використання скриптів, що є його недоліком.

Ще одним недоліком є відносно великий розмір додатків.

Один з плюсів Юніті — це список підтримуваних платформ, на яких може запускатися програма. Unity працює майже всюди - на ПК (всі операційні системи), на Андроїд, на iOS, на SmartTV, в браузері, на різних екзотичних системах — наприклад, Tizen OS. [25]

Потужний плюс Юніті — це ассети. Все в програмі, включаючи код, картинки, представляється ассетами (Asset). Ассет можна експортувати, імпортувати. Таким чином, сторонні розробники можуть робити цілі заготовки для ігор. Все, що залишиться — це замінити картинки і підправити скрипти.

## **4.2 AutoCAD**

AutoCAD - це програма для автоматизованого проектування (CAD), яка використовується для дво- та тривимірного проектування та складання. AutoCAD розроблений та проданий компанією Autodesk Inc. і був однією з перших програм САПР, яку можна було виконати на персональних комп'ютерах.

Спочатку AutoCAD був похідний від програми під назвою Interact, яка була написана власною мовою. Перший випуск програмного забезпечення використовував лише примітивні об'єкти, такі як багатокутники, кола, лінії, дуги та текст для побудови складних об'єктів. Пізніше він став підтримувати власні об'єкти через інтерфейс програмування додатків C ++.

Сучасна версія програмного забезпечення включає повний набір інструментів для твердого моделювання та 3-D. AutoCAD також підтримує численні інтерфейси прикладних програм для автоматизації та налаштування. DWG (малюнок) – це власний формат файлу для AutoCAD і основний стандарт для сумісності даних CAD. Програмне забезпечення також надає підтримку для дизайну веб-формату (DWF), формату, розробленого Autodesk для публікації даних CAD.

AutoCAD може створити будь-який двовимірний малюнок та 3D-модель або конструкцію, які можна намалювати вручну. Програма також дозволяє користувачеві групувати або шарувати об'єкти, зберігати об'єкти в базі даних для подальшого використання та маніпулювати властивостями об'єктів, такими як розмір, форма та розташування. AutoCAD має численні програми у широкому спектрі областей.

Програму можна використовувати для простих проектів, таких як графіки чи презентації, або складних конструкцій, наприклад, для складання архітектури будівлі. Деякі інші практичні програми можуть включати:

- дизайн інтер'єру;
- аеронавігаційні конструкції;
- логотипи;
- образотворче мистецтво;
- карти;
- вітальні листівки;

- інженерні конструкції;
- архітектурні проекти.

### **4.3 Інтегроване середовище розробки Microsoft Visual Studio**

Visual Studio , також відомий як Microsoft Visual Studio і VS - це інтегроване середовище для розробки програм, використовуваним Microsoft Windows. Це інструмент для написання комп'ютерних програм , веб-сайтів , веб-додатків та веб-служб . Він включає редактор коду , налагоджувач , інструмент проектування графічного інтерфейсу та конструктор схем бази даних та підтримує більшість основних систем контролю версій. Він доступний як у безкоштовній "Громадській" редакції, так і в платній комерційній версії.

Visual Studio підтримує різні мови програмування та дозволяє редактору коду та налагоджувачу підтримувати майже будь-яку мову програмування, за умови існування послуги, що залежить від мови. Вбудовані мови включають C, C ++ та C ++ / CLI, Visual Basic .NET, C # і F #. Підтримка інших мов, таких як Python, Ruby, Node.js та M серед інших, доступна через мовні служби, встановлені окремо. Він також підтримує XML / XSLT, HTML / XHTML, JavaScript та CSS. Java підтримувались і раніше. [26]

Visual Studio Microsoft - це IDE (комплексне середовище розробки) для мов програмування Microsoft .NET, таких як VB.NET Shop та C #. Я здивований, що він є на вашому ПК, якщо ви його не встановили, але сказали, що якщо ви не займаєтесь розробкою будь-якої програми, її можна видалити.

Існує версія Mac, але вона походить від студії Xamarin, яка сама походить від MonoDevelop, який був портом інструменту відкритого коду з відкритим кодом (shardevelop) с # deveoplment від .net.

### **4.4 Мова програмування C#**

C # - це загальна об'єктно-орієнтована програма програмування (ООП) для мереж та веб-розробок. C # визначається як загальна мовна інфраструктура (CLI). У січні 1999 року голландський інженер програмного забезпечення Андерс Хейльсберг створив команду з розробки C # як доповнення до NET-системи Microsoft. Спочатку C # був розроблений як C-подібний об'єктно-орієнтований мова (Cool). Власне ім'я було змінено, щоб запобігти потенційним проблемам із торговою маркою. У січні 2000 року NET був випущений як C #. [27]

Її структура NET просуває безліч веб-технологій. Термін іноді пишеться як C Sharp або C-Sharp. Символ # терміна походить від музичної гострої клавіші, що позначає збільшення на один півтоновий крок. C # вимовляється "бачити гострий".

C # покращив та оновив багато функцій C та C ++, зокрема: C # має суворий тип змінної булевих даних, такий як bool, тоді як типи змінних bool C ++ можуть повертатися у вигляді цілих чи покажчиків, щоб уникнути поширених помилок програмування. C # автоматично управляє недоступною об'єктом пам'яті за допомогою сміттєзбірника, що усуває проблеми розробника та витрати пам'яті.

Тип C # безпечніший, ніж C ++ і має лише безпечні конверсії за замовчуванням (наприклад, розширення цілих чисел), які реалізуються під час компіляції або виконання. Жодні неявні перетворення між булевими, членами перерахування та цілими числами (крім 0) не можуть бути перетворені в перелічений тип. Конверсії, визначені користувачем, повинні бути вказані як явні або неявні, порівняно з операторами неявного перетворення за замовчуванням C ++ та конструкторами копіювання.

## **4.5 Платформа ARCore**

Корпорація Google представила 29 серпня платформу ARCore - вона дозволить створювати додатки з доповненою реальністю під Android (аналогічну платформу для iOS Apple представила в червні). Поки ARCore працює тільки на двох моделях

телефонів, але Google сподівається, що в майбутньому додатки з доповненою реальністю зароблять на мільйонах пристроїв. [28]

ARCore - це набір засобів для розробки програм (SDK) для операційної системи Android. Розробники можуть використовувати ARCore при створенні своїх програм для доповненої реальності. ARCore визначає горизонтальні поверхні і стежить за рухом камери. Ця платформа заснована на трьох ключових технологіях: розуміння навколишнього світу (щоб знаходити горизонтальні поверхні і визначати їх розмір); відстеження руху (щоб віртуальний об'єкт «стояв» на цій поверхні, незважаючи на тряску телефону); оцінка освітлення (щоб віртуальний об'єкт міг реагувати на реальний світ - відкидати тіні або лякатися в темряві) (рисунок 4.1).



Рисунок 4.1 – Застосування ARCore для доповненої реальності

ARCore заснована на проєкті Tango. У 2014 році Google представила платформу для доповненої реальності Tango. Її особливістю було те, що в ній використовувалися спеціальні телефони та планшети з додатковими сенсорами для кращого розуміння простору. Tango так і не привернула уваги розробників, і її можна вважати мертвим проєктом. ARCore працює на звичайних телефонах. У цьому головна відмінність від Tango: для ARCore не потрібні додаткові сенсори, досить тих, що вже є (в першу чергу камери і гіроскопа). Завдяки цьому Google сподівається, що додатки на базі нової



платформи скоро запрацюють на 100 мільйонах пристроїв. Поки, правда, підтримуються тільки два телефони - Google Pixel / Pixel XL і Samsung Galaxy S8. [29]

Як відзначають в Google, в протягом останніх трьох років компанія працювала над технологіями, які сприяють розвитку доповненої реальності на мобільних пристроях, за допомогою платформи Tango. Вона стала підґрунтям для створення ARCore. Завдяки тому, що цей SDK не вимагає додаткового обладнання, він може застосовуватися на різних пристроях Android. У Google планують запуснути ARCore на мільйони пристроїв. Він доступний на Pixel і Samsung S8, які працюють на Android 7.0 і пізніших версіях. До завершення тестового періоду пошуковий гігант розраховує, що ARCore буде працювати на 100 млн пристроїв. Для того щоб він працював якісно і приносив хороші результати, Google співпрацює з Samsung, Huawei, LG, ASUS і іншими компаніями. [30]

ARCore працює на Java / OpenGL, Unity і Unreal і фокусується на таких напрямках:

- відстеження руху. Використовуючи камеру телефону для відстеження опорних точок в кімнаті (п.п. ці точки визначають місце, де буде розташований віртуальний об'єкт) й даних гіроскопа, ARCore визначає положення і орієнтацію пристрою під час руху. При цьому віртуальні об'єкти залишаються саме там, де їх розмістили;
- розпізнавання навколишнього середовища. Зазвичай об'єкти доповненої реальності розміщуються на підлозі або столі. ARCore може розпізнавати горизонтальні поверхні, використовуючи ті ж опорні точки, що і при відстежуванні руху;
- оцінка освітлення. ARCore визначає рівень освітленості навколишнього середовища і дає можливість розробникам висвітлювати віртуальні об'єкти відповідно до обстановки навколо. Завдяки цьому вони виглядають ще більш реалістично. [31]

## **Висновки до розділу 4**

В даному розділі були описані засоби розробки, що використовуються для налаштування та створення підсистеми тривимірної інтеграції об'єкту, а також обгрунтовано вибір вказаних засобів.

Для розробки підсистеми використаний Unity – це ігровий рушій, що є сумісним з фреймворком ARCore та технологією ARKit, що використовується в цілісній системі та надає можливість використання в додатку доповненої реальності й надає більші можливості при використанні додатку користувачем.

Програмний код був написаний мовою C#, що підтримує об'єктно-орієнтований підхід.

Для побудови початкової схеми та запису метаданих було використано програму AutoCAD, що є безкоштовною та надає всі необхідні можливості для задання схемі точності, що є дуже важливим при використанні додатку.

## 5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

При запуску додатку, користувач повинен визначити позицію відносно карті на додатку за допомогою маркеру. Потрібно навести камеру на маркер на стіні і за допомогою порівняння унікального ідентифікатора додаток визначає положення користувача відносно карти (рисунок 5.5)

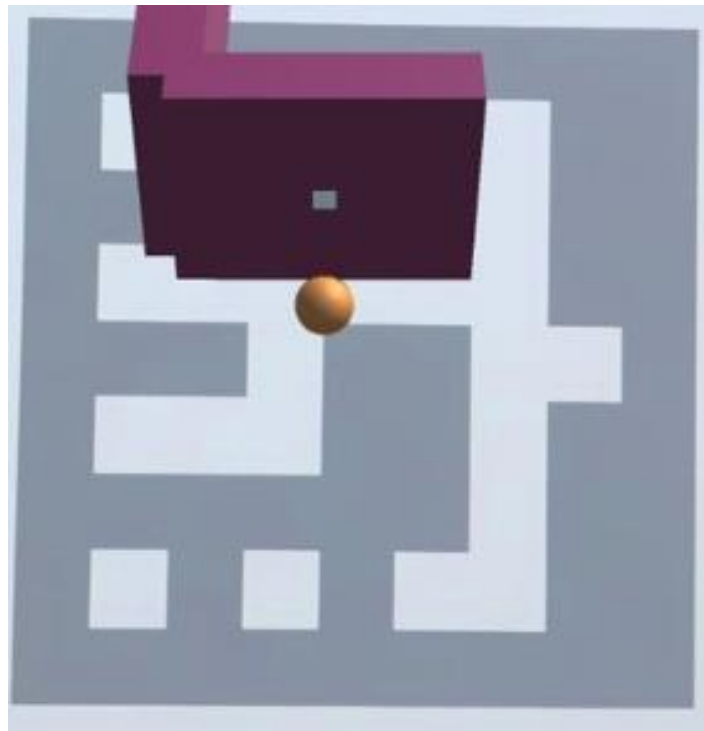


Рисунок 5.5 – Положення користувача на карті після зчитування маркеру

Потім користувачу потрібно обрати з випадаючого списку пункт призначення, номер потрібного кабінету. Альтернативою є встановлення мітки на карті вручну, але ефективішим буде обрати один із запропонованих пунктів призначення (рисунок 5.6).

Якщо пункт призначення знаходиться на іншому поверсі – користувача додаток приведе до найближчих сходів, що можуть вести на потрібний поверх. Після переходу на потрібний поверх, користувачу потрібно знову синхронізуватися з маркером.

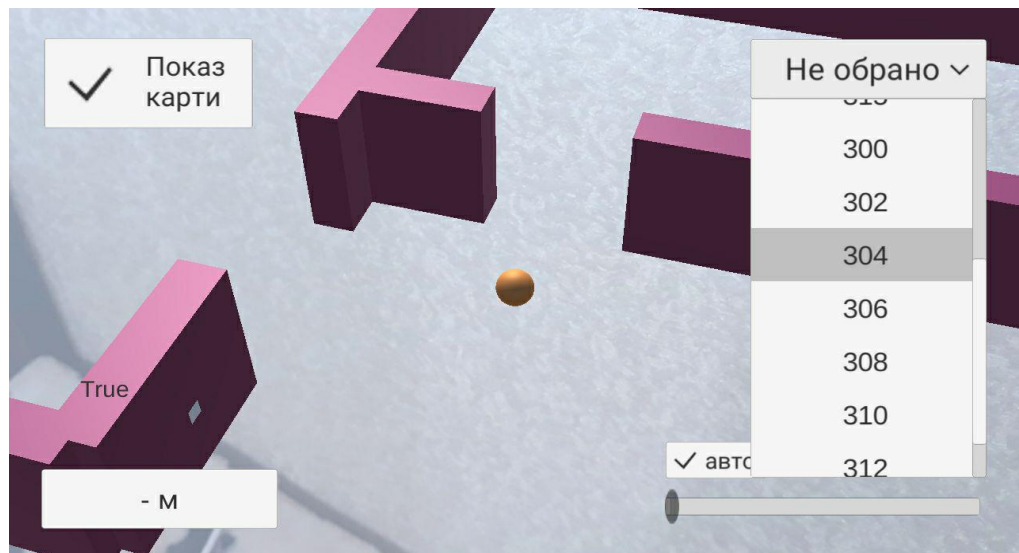


Рисунок 5.6 – Вибір пункту призначення з випадального списку

При проходженні користувача і поверненні карти – схема також повертається у відповідності з використанням кватерніонів, що було описано в дипломній дисертації. Також, карту можна повертати під зручним кутом самостійно, використовуючи налаштування в нижньому правому куті карти (рисунок 5.7).

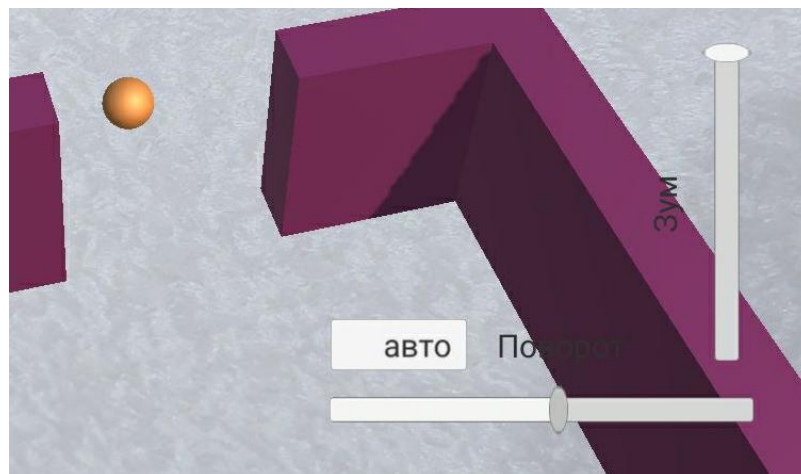


Рисунок 5.7 – Поворот карти

Можлива зміна масштабу карти, при цьому схематичне відображення карти не буде зіпсоване (рисунок 5.8).

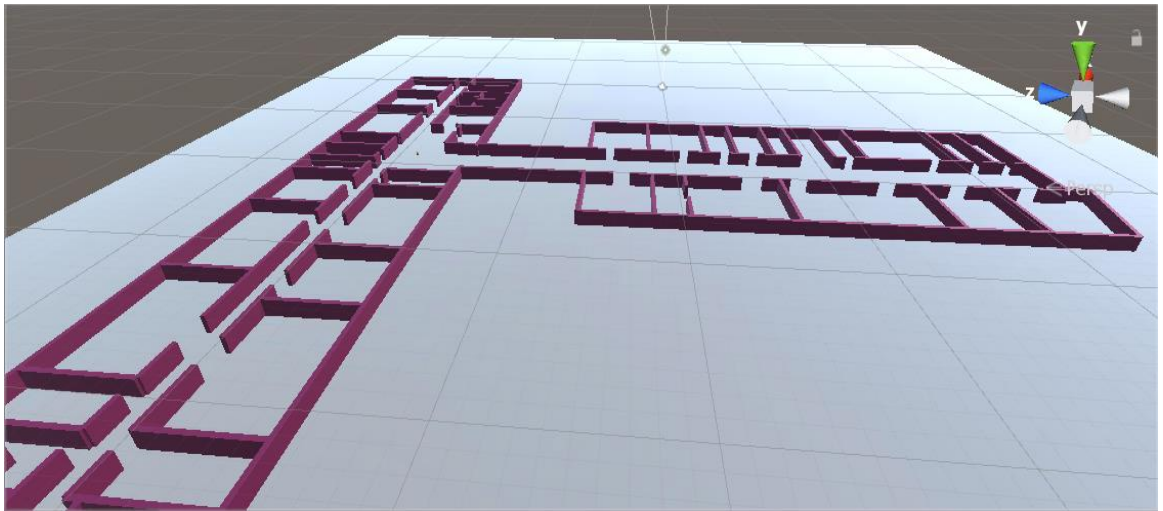


Рисунок 5.7 – Масштабування карти

Також можливий режим камери з використанням доповненої реальності (рисунок 5.1).

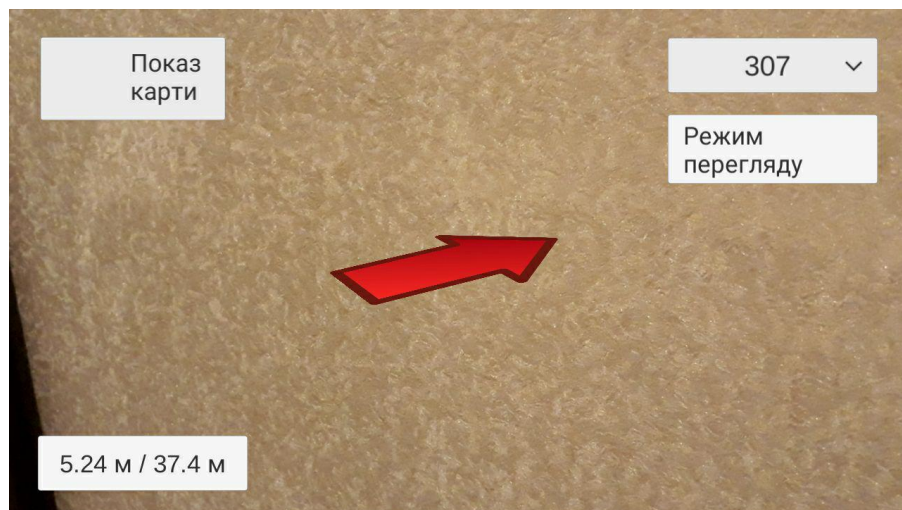


Рисунок 5.8 – Режим камери з використанням доповненої реальності.

Після закінчення проведеного маршруту, користувач отримає повідомлення про досягнення точки призначення.

## **Висновки до розділу 5**

В розділі була наведена детальна інструкція роботи користувача з системою. Описані дії та маніпуляції, що користувач може здійснювати з картою при використанні додатку, а також вказані дії, що має зробити користувач для встановлення точки позиції на карті.

Додаток має зрозумілий інтерфейс, а також впливаючі повідомлення з інструкціями для подальших дій користувача, наприклад при переході між поверхами для досягнення точки призначення. Це має мінімізувати будь-які складнощі роботи користувача з додатком.

## 6 РОЗРОБКА СТАРТАП ПРОЕКТУ

Розроблювану систему можна вважати стартапом, тому що використане вирішення для навігації в будівлі є новим, не вимагає додаткових затрат, крім тих, що може дозволити собі звичайний користувач, адже мобільні телефони на сьогоднішній день є у кожного.

Додаток не потребує значних ресурсів, лише підтримки технології ARKit та камери на мобільному телефоні, то ж для цього не потрібен найновіший смартфон.

Актуальність додатку та його новизна з конкурентною спроможністю буде розглянута та обґрунтована в даному розділі. Було розглянуто основних конкурентів на ринку та проведено оцінку

### 6.1 Опис ідеї проекту

Будь-який проект починається з ідеї. Розробка даної системи — не виняток. Саме тому у таблиці 6.1 описана ідея розробки стартап-проекту.

Таблиця 6.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Додаток навігації всередині приміщення для смартфонів з використанням доповненої реальності	1. Навігація в навчальних корпусах університетів, шкіл і т.п.	Швидке знаходження необхідної аудиторії; особливо актуально для нових студентів або учнів.

Таблиця 6.1 (продовження)

	2. Торгові центри та магазини	Дозволяє зекономити час і дістатися до необхідного відділу магазину чи товару швидше; для власника магазину (непрямий користувач) це означатиме збільшення потоку користувачів, а також введення реклами товарів за бажанням.
--	-------------------------------	---

З наведеної таблиці виділені напрямки застосування додатку. Це можуть бути не лише навчальні заклади, а й великі торгівельні центри зі складною структурою. Звичайно, це вже буде комерційне використання, що буде приносити дохід.

В навчальних закладах є частою проблема знаходження потрібного кабінету, особливо якщо є ділення на корпуси. Додаток допоможе знайти найкоротший шлях користувачеві.

Наступним кроком є визначення характеристик продукту, що планується розробити.

Таблиця 6.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п / п	Техніко - економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Розроблюваний проект	Конкурент 1	Конкурент 2	Конкурент 3			
1	Точність	середня	висока	висока	низька		+	
2	Собівартість	низька	середня	висока	висока			+



Таблиця 6.2 (продовження)

3	Використання додаткового обладнання	Ні	Ні	Так	Так			+
4	Кросплатформенність	Ні	Так	Так	Так	+		
5	Використання доповненої реальності	Так	Так	Ні	Ні			+

З таблиці вище зрозуміло, що відбувається порівняння з деякими конкурентами. За продукти конкурентів було обрано ті, що і в розділі 2 в пункті 2.2. Конкурент 1 — Inplaces, 2 — Hubbell IPS, 3 — IndoorAtlas.

Проаналізувавши таблицю, зрозуміло, що актуальність додатку значна, так як він не використовує додаткового обладнання, що часто може дорого коштувати. Запропонований же додаток надає можливість навігації без застосування додаткових технологій, потрібен лише телефон, навіть не найновіший. До того ж, точність навігації додатку до 10 см. Звичайно, похибки можливі, технологія не досконала і потребує доопрацювань.

На основі цих міркувань можна стверджувати конкурентноспроможність продукту.

## 6.2 Технологічний аудит ідеї проекту

В цьому пункті слід провести аудит технології проекту, що має допомогти реалізувати ідею (технології створення товару).

Перший крок для цього — це визначення технологій, які можуть використовуватися в процесі розробки (таблиця 6.3).

Таблиця 6.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Елементи доповненої реальності	ARCore	Наявна	Доступна
		ARKit	Наявна	Доступна
		Wikitude	Наявна	Платна
2	Мітки ля навігації по карті	ArUCO	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: ARCore, ArUCO				

З наведеної таблиці видно, що для додання в проект елементів доповненої реальності можуть бути використані кілька фреймворків. ARCore дозволяє розгортати додатки на пристроях, що працюють на базі Android, але також є ймовірність, що програма запуститься і на iOS. ARKit — це технологія Apple, розроблена спеціально під iOS. Wikitude — ще AR фреймворк, проте він платний, і його обрання збільшить собівартість продукту. При цьому якість та можливості ARCore не поступаються Wikitude. До того ж, він безкоштовний. Саме тому з трьох аналогів було обрано ARCore.

Для реалізації створення схеми будівлі було використано безкоштовний застосунок AutoCAD, для встановлення міток та точок ArUCO

### 6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. [2]

Спочатку варто провести аналіз попиту (таблиця 6.4).

Таблиця 6.4 — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Біля 5
2	Загальний обсяг продаж, грн/ум.од	25 000 грн за ум.од.
3	Динаміка ринку (якісна оцінка)	Очікується зростання
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	15

Переглянувши таблицю, можна дійти висновків, що проект є перспективним, а капіталовкладення окупиться, так як дана сфера ще розвивається і є актуальною.

Наступним кроком є визначення цільової аудиторії програмного продукту, її поведінки та вимог (таблиця 6.5).

Таблиця 6.5 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

Таблиця 6.5 (продовження)

1	Можливість швидко і зручно здійснювати навігацію будівлею	Учні, студенти	Не передбачається	Зручність інтерфейсу, швидкість роботи, можливість швидко дістатися до аудиторії
2		Покупці в торгових центрах та схожих структурах	Не передбачається	Зручність інтерфейсу, швидкість роботи, можливість швидко дістатися до місця призначення, достатньо велика точність (через те, що будівлі подібного характеру зазвичай більші)

Наступним кроком є аналіз ринкового середовища. Необхідно в табличному вигляді визначити фактори загроз (таблиця 6.6) і можливостей (таблиця 6.7), що перешкоджають і допомагають впровадженню програми. [32]

Таблиця 6.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Ризик технології	Навігація без використання додаткового обладнання згодом може стати недостатньо точною для певного прошарку клієнтів	Підвищення точності за рахунок введення якого одного типу додаткових засобів (наприклад, Wi-Fi)

Таблиця 6.6 (продовження)

2	Недоступність платформи	Неможливість використання програмного забезпечення на пристроях, що не запускаються на базі андроїд	Розробка програмного забезпечення для інших платформ (наприклад, iOS).
---	-------------------------	---	--

Переглянувши основні загрози, що наведені в попередній таблиці, стає очевидним, що технологія не є досконалою і потребує доопрацювання. Звичайно, додаткові технології можуть надавати більшої точності, але в даному випадку необхідним є стабільний Wi-Fi сигнал, що не завжди є можливим.

Також загрозою є можливість розгортання на інших платформах, таких як iOS, що може не підтримувати ARCore.

Таблиця 6.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Будування шкіл, вищих навчальних закладів, торгових центрів або розширення існуючих	Існує можливість, що при збільшенні потенційних покупців (структур, яким може знадобитися продукт), збільшить і попит на забезпечення	Посилення рекламної кампанії

Найголовнішим фактором можливостей є необхідність навігації по великих будівлях, таких як бізнес-центри, що орендує одночасно велика кількість організацій та підприємств. Також торгові центри з великою кількістю магазинів, та навчальні заклади.

Наступним кроком є визначення типу конкуренції (Таблиця 6.8)

Таблиця 6.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика
Тип конкуренції	Олігополія
За рівнем конкурентної боротьби	Глобальна
За галузевою ознакою	Міжгалузева
Конкуренція за видами товарів	Товарно-видова
За характером конкурентних переваг	Нецінова
За інтенсивністю	Не марочна

Такий аналіз конкуренції є непоганим, але неповним. Саме тому варто провести аналіз конкуренції в галузі за М. Портером (таблиця 6.9).

Таблиця 6.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
------------------	---------------------------	-----------------------	---------------	---------	------------------

Таблиця 6.9 (продовження)

	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	Існує близько 5 найвпливовіших конкурентів. Конкурент 1 (Inplaces) є найбільш серйозним, адже використовує схожий підхід і технології	Можливості виходу на ринок є, але вони також є у потенційних конкурентів. Строки виходу їх продукту на ринок — приблизно 1 рік	Постачальники відсутні	Зручність, достатня точність	Загроза розробки конкурентами більш точного та (або) дешевшого аналогу

Використовуючи результат попередньої таблиці, стає можливим виділити основні фактори конкурентоспроможності продукту (Таблиця 6.10).

Таблиця 6.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Функціонал	Наявність режиму з доповненою реальністю, що додає цікавості і зручності у використанні, а також режиму перегляду карти

Таблиця 6.10 (продовження)

2	Вартість	Передбачається випуск товару з ціною, нижчою ніж у конкурентів
3	Не потребує додаткового обладнання	Для роботи програми не потрібні, наприклад, Bluetooth beacons

Таким чином, низька собівартість програмного продукту (лише смартфон, що є у кожного на сьогоднішній день), а також те, що додаток не використовує додатковк обладнання, можна зробити висновок, що додаток є конкурентоспроможним.

Після таких висновків вже можна проаналізувати сильні та слабкі сторони проекту у порівнянні з конкурентами (таблиця 6.11).

Таблиця 6.11 — Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним продуктом						
			-3	-2	-1	0	1	2	3
1	Функціонал	17			+				
2	Вартість	19	+						
3	Не потребує дод. обладнання	17			+				

Заключною таблицею ринкового аналізу можливостей впровадження розробленого програмного продукту є аналіз SWOT, що представляє собою сильні та слабкі сторони, можливості і загрози для проекту в одній таблиці (таблиця 7.12).



Таблиця 6.12 — SWOT-аналіз стартап-проекту

Сильні сторони: функціонал, вартість, не потребує дод. обладнання	Слабкі сторони: Недоступність на платформах, що не підтримують ARCore, недостатня точність для деяких клієнтів
<p>Можливості:</p> <p>Продаж більшої кількості товару через збільшення навчальних закладів, торгових центрів</p> <p>Індивідуалізація програми під конкретного замовника</p> <p>Гібридизація технологій для підвищення точності</p>	<p>Загрози: віддання переваги потенціальними клієнтами продуктам конкурентів через недоступність інших платформ та технологічний ризик (не використовуються дод. апаратура)</p>

Використовуючи SWOT-аналіз, можна скласти таблицю 6.13 альтернатив ринкового впровадження стартап-проекту.

Таблиця 6.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка системи навігації з використання додаткового обладнання (Bluetooth Beacons)	20%	1,5
2.	Розробка системи навігації з використанням тільки комп'ютерного зору	90%	1

Проаналізувавши результати таблиці, очевидно, що найбільш вигідною альтернативою є розробка системи навігації з використанням тільки комп'ютерного зору, так як вона не потребує інтеграції з додатковим обладнанням.

## 6.4 Розробка ринкової стратегії проекту

В розробленні ринкової стратегії перше, що треба зробити, це визначити стратегію охоплення ринку (таблиця 6.14).

Таблиця 6.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Навчальні заклади	+	Середній	Практично немає	Просто
2	Торгові центри	+	Високий	Вище середнього	Вище середнього
3	Великі людні установи, наприклад, аеропорти	+	Вище середнього	Висока	Складно
Які цільові групи обрано: 1, 2					

Цільовою аудиторією є навчальні заклади, бізнес-центри та торгові центри, так як вони потребують навігації та вказання місцезнаходження.

Звичайно, великі торгові центри потребуватимуть довготривалої попередньої підготовки схеми будівлі та встановлення маркерів, але це необхідно для надання додатку необхідної точності (до 10 см) та виключення використання додаткових платних технологій та обладнання.

На основі цих міркувань можна навести базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробка системи навігації з використанням тільки комп'ютерного зору	Стратегія диференційованого маркетингу	Функціонал, Вартість, Не потребує додаткового обладнання	диференціація

Так як планується співпраця з кількома сегментами ринку, за стратегію охоплення ринку було обрано стратегію диференційованого маркетингу; до того ж, деякі особливості продукту будуть змінюватись в залежності від сегменту і конкретного користувача. Базовою стратегією розвитку визначено диференціацію, адже товар пропонує не зовсім поширений підхід: доповнену реальність і відмову від додаткового обладнання.

Наступний крок — визначення базової стратегії конкурентної поведінки (таблиця 6.16)

Таблиця 6.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект "першопрохідцем" на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Шукати нових споживачів	Так	Стратегія наслідування лідеру

Зважаючи на вимоги користувачів, стратегії розвитку та конкурентної поведінки, далі треба розробити стратегію позиціонування. Під цією стратегією розуміється формування ринкової позиції (комплексу асоціацій), що допоможе споживачам з ідентифікацією проекту (таблиця 6.17).

Таблиця 6.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
----------	--	---------------------------------	---	--

Таблиця 6.17 (продовження)

1	Дешевизна, зручність інтерфейсу, точність	Диференціація	Відмова від додаткового обладнання зробить продукт дешевшим, а режим з доповненою реальністю допоможе цікавіше і швидше знаходити бажане місце	Доступність, зручність, функціональність
---	---	---------------	--	--

Отже, додаток є зручним, безкоштовним та доволі простим у використанні, що робить його доступним для звичайного користувача.

## 6.5 Розроблення маркетингової програми

Розробляючи маркетингову програму, з самого початку потрібно сформулювати маркетингову концепцію товару (таблиця 6.18).

Таблиця 6.18 — Визначення ключових переваг концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Функціонал	Режим доповненої реальності	Наявність режиму, якого практично немає ні в одного з конкурентів
2	Вартість	Вигідна ціна для покупців	Нижча ціна, ніж у більшості конкурентів

Таблиця 6.18 (продовження)

3	Незалежність системи від дод. пристроїв	Не потребує додаткового обладнання	Потрібен тільки смартфон
---	---	------------------------------------	--------------------------

Наступний крок — розробка трирівневої маркетингової моделі товару.

Таблиця 6.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Додаток для смартфона, що допомагає здійснювати навігацію всередині будівлі, використовуючи доповнену реальність		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Низька ціна 2. Функціонал 3. Не потребує дод. обл. 4. Зручний інтерфейс	Нм	Тх
	Якість: продукт протестовано вручну		
	Маркування: Немає		
	Компанія: назва “Tescinnav”		
III. Товар із підкріпленням	Продаж товару за підпискою; постійна підтримка придбаного додатку		
За рахунок чого потенційний товар буде захищено від копіювання: обфускація коду			

Далі потрібно визначити цінові межі (таблиця 6.20).

Таблиця 6.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
1.	30000	40000	350000	20000-25000

Після цього необхідно визначити систему збуту, оптимальну для даного проекту (таблиця 6.21).

Таблиця 6.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Покупка та щомісячна підписка	Продаж	0 (напрямую)	Власна

Попередня таблиця презентує систему збуту. В її рамках прийняті такі рішення, що користувачі купуватимуть щомісячну підписку і що система збуту буде без посередників, власна, тобто глибина каналу збуту — 0.

Завершальним етапом формування маркетингової програми є розробка концепцій маркетингових комунікацій (таблиця 6.22).

Таблиця 6.22 — Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цілових клієнтів	Канали комунікацій, якими користуються цілові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1.	Завантаження додатку через QR-коду, розміщеного в приміщенні.	Інтернет	Низька ціна, функціонал, не потрібно додаткового обладнання, зручний інтерфейс	Зацікавити потенційних користувачів; довести, що розроблений додаток економить час та допомагає спростити процес навігації.

Посилаючись на попередню таблицю, що показує концепцію маркетингових комунікацій для розробленого програмного продукту, потрібно вказувати QR-код для можливості завантажити додаток і використання його для навігації.

Не зайвим також буде додати аналіз ефективності проекту. Далі він буде наведений на двох рисунках (рисунок 6.1, рисунок 6.2).



Постійні	Змінні
Оренда приміщення - 20000	Витрати на збут – 2% - 500
Ком. послуги – 5000	<u>Хостинг</u> – 10% - 2600
<u>Заробітня плата</u> – 100000	
Маркетинг – 100000	Ремонт і обслуговування техніки – до 20000
Податки - 19500	
<u>Всього</u> : 244500	

Вартість ліцензії: \$1000/міс + опціональні додаткові функції

СЗВ = \$120=3200

ТВ = 244500/(26000 – 120 \* 26)=11 ліцензій

Рисунок 6.1 — Витрати на проект

З рисунку вище ми можемо побачити обраховані витрати на проект за рік. До постійних витрат слід віднести ті, які стабільно будуть витрачатися протягом року (напр., оренда приміщення або податки). Змінні витрати — це витрати, які можуть змінюватися час від часу (ремонт техніки). Також було пораховано вартість однієї ліцензії, середні змінні витрати та точку беззбитковості, що дорівнює 11 ліцензіям.

• Ставка дисконтування: 40%/рік

Показник	0 рік	1 кв	2 кв	3 кв.	4 кв
1. Сума інвестицій, тис. грн.	-600	-	-	-	-
2. Виручка від реалізації, тис. грн.	-	260	875	1375	2250
3. Витрати на експлуатацію проекту, тис. грн.	-	766	840	900	1005
5. Ставка дисконту, %	-	10	10	10	10
6. Коеф. дисконтування	-	0.91	0.83	0.75	0.68
6. Грошові потоки, тис. грн.	-	-506	35	475	1245
7. Дисконтовані грошові потоки, тис. грн.	-	-460	29	356	847
8. NPV	-	-1060	-1031	-696	151

$$k_d = \frac{1}{(1 + d)^t}$$

$$TO = 3\text{кв} + \frac{696}{847}\text{кв}$$

$$3.82 \text{ кв} = 11.5 \text{ місяців}$$

$$BCR = \frac{3524}{2752} = 1.28$$

$$P_c = \frac{BCR - 1}{t} = \frac{0.28}{4} = 7\%$$

Рисунок 6.2 — Аналіз ефективності

З аналізу ефективності видно, що витрати на створення додатку мають повністю окупитися через 11,5 місяців.

## **Висновки до розділу 6**

В даному розділі було проведено аналіз конкурентоспроможності та оцінка окупності запропонованого додатку. Оцінка стратегії входу на ринок, а також можливі ризики та методи вирішення за допомогою SWOT-аналізу.

Для реалізації було вирішено обрати технологію доповненої реальності та комп'ютерного зору, безкоштовних технологій. Звичайно, додаток потребує доопрацювання для масштабного використання, але використання безкоштовних технологій, що не потребують значного доопрацювання дуже вирізняє додаток поміж конкурентів.

Також, додаток конкурує за точністю з конкуруючими технологіями, наведений аналіз та порівняння з Inplaces, Hubbell IPS та IndoorAtlas. Проведений аналіз дає можливість зробити висновок, що конкуренти мають більшу можливість використання автономно, тобто з використанням автопілотів та роботів. Але цінова політика додаткового обладнання, що вони використовують надає можливість порівняння з розробленим додатком та позитивно вирізнитися.

Розглянута можливість комерціалізації додатку для використання не лише в навчальних закладах, а й торгових центрах, великих магазинах, музеях. Для цього потрібні невеликі доопрацювання, такому як налаштування схеми вхідних даних.

# ВИСНОВКИ

Під час виконання роботи було проведено дослідження існуючих систем навігації, результатом яких є висновки про недосконалість та недостатню точність побудови карти будівлі для встановлення точної позиції користувача і проведення маршруту. Наслідком дослідження було запропоновано створити такий додаток, що може надавати прийнятну точність встановлення позиції та проведенні маршруту з використанням обладнання, наявного в кожного – смартфон.

Було створено підсистему інтеграції тривимірної моделі об'єкту, що є частиною мобільного додатку для навігації корпусом, містить в собі необхідні метадані та маркери, що потрібні для коректного відпрацювання інших модулів системи. Також попередньо було проаналізовано проблеми створення даної підсистеми.

Підсистема надає такі можливості:

- генерація тривимірної моделі об'єкту карти на сцені додатку;
- зчитування попередньо записаних метаданих для вибору точки призначення;
- формування та відображення маркера для встановлення початкової позиції користувача на карті;
- можливість масштабувати карту та повертати під потрібним кутом без критичної зміни відображення карти;
- можливість завантаження та інтеграції інших попередньо підготованих схем будівель для подальшого використання;
- інші додаткові опції.

Програмний продукт було протестовано. Під час цієї перевірки він показав достатню точність для навігації з точки А в точку В з використанням лише камери мобільного телефону та доступу до мережі Інтернет, коректному зчитуванню даних для встановлення позиції користувача та високу точність прокладання маршруту завдяки відповідному зчитуванню масштабу схеми та її відрисовування в додатку.

Для розробки програмного додатку було обрано крос-платформений інструмент Unity, що підтримує необхідну бібліотеку для інтеграції двовимірної схеми будівлі в тривимірну карту, з об'єктами, що можна розпізнати та визначити

кватерніон для повороту карти. Мовою розробки обрано C#, що підтримує об'єктно-орієнтований підхід. Середовищем розробки обрано Microsoft Visual Studio. Для попередньої підготовки схеми, що інтегрується в додаток, було обрано AutoCAD, що надає можливість запису метаданих для створення точок призначення та генерації маркеру.

Додаток є актуальним для прокладання маршруту в невідомій будівлі та може бути використаний для комерційних проектів. Було проведено дослідження існуючих рішень та знайдено оптимальне рішення з використанням обмежених ресурсів та дешевизни. Додаток має зрозумілий інтерфейс та підказки користувачу.

Додаток має потенціал для комерційного використання не лише в навчальних закладах, а й в інших сферах. Звичайно, обмежувальним фактором є створення попередньої схеми та внесення метаданих з використанням сторонніх програм для інтеграції тривимірного об'єкту, але поки не відомі аналогічні рішення, що будуть надавати такої ж точності та можливостей внесення додаткових даних. Тому програмний додаток має потенціал для розвитку та вдосконалення.

## СПИСОК ЛІТЕРАТУРИ

1. Компас или AutoCad [Електронний ресурс] – Режим доступу до ресурсу: [https://cl-box.ru/kompas\\_vs\\_autocad](https://cl-box.ru/kompas_vs_autocad).
2. Soldatov D. Mono visual simultaneous localisation and mapping for autonomous robots on TRIK / D. Soldatov. // Saint-Petersburg – 2016.
3. Prasad R. Applied satellite navigation using GPS, GALILEO, and augmentation systems / R. Prasad, M. Ruggieri. // Boston: Artech — 2005.
4. Gu Y. A survey of indoor positioning systems for wireless personal networks / Y. Gu, A. Lo, I. Niemegeers. // IEEE Communications Surveys & Tutorials. – 2009. – №11. – С. 13–32.
5. Beis J.S., Lowe D.G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. / Beis J.S., Lowe D.G. – 2008.
6. Dailey M.N., Parnichkun M. Landmark-based simultaneous localization and mapping with stereo vision. / Dailey M.N., Parnichkun M. – 2011.
7. Brian X. Chen. If You're Not Seeing Data, You're Not Seeing . – 2009.
8. ARCore - Google Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar>.
9. Робототехника. / Фу К., Гонсалес Р., Лі К. // М.: Мир – 1989. – С. 622.
10. Bay H., Tuytelaars T., Van Gool L. SURF: speeded-up robust features. / Bay H., Tuytelaars T., Van Gool L. – 2015
11. Terryberry T. B., French L. M., Helmsen J. GPU accelerating speeded-up robust features. / Terryberry T. B., French L. M., Helmsen J. – 2010
12. Implementing the scale invariant feature transform (SIFT) method. / Meng Y., Dr. Tiddeman B. – 2010.
13. XDATA AutoCAD [Електронний ресурс] – Режим доступу до ресурсу: <https://knowledge.autodesk.com/ru/support/autocad/learn-explore/caas>
14. Xiang Zhang, Stephan Franz, Nassir Navab. Visual Marker Detection and Decoding in AR Systems: A Comparative Study. / Xiang Zhang, Stephan Franz, Nassir Navab — 2016

15. Mehmet Sezgin. Survey over image thresholding techniques and quantitative performance evaluation. / Mehmet Sezgin — 2017
16. Ehsan Nadernejad. Edge Detection Techniques: Evaluations and Comparisons. / Ehsan Nadernejad — 2017
17. Study E. Ueber Nicht-Euklidische und Liniengeometrie. /Festschr. der Philos. // Fakult. zu Greifswald — 1900.
18. Теорія винтів і комплексні числа. / Котельніков А.П. //Гостехиздат — 1950.
19. Базиси в евклідових просторах і ряди Фур'є / Ільїн В.А. // Соросовский Освітній Журнал — 1998. № 4. С. 95 – 101.
20. Lectures on Quaternions. / Hamilton W.R. // Dublin — 1853.
21. Preliminary Scetch of Biquaternions./ Clifford W. // Proc. of London Math. Soc., — 1873. — v. IV. — С. 381-393.
22. Johnson B. Professional Visual Studio 2017 / Bruce Johnson. – Indianapolis: John Wiley & Sons – 2017.
23. Goldstone W. Unity 3.x Game Development Essentials / W. Goldstone. // Birmingham, Mumbai: Packt Publishing — 2009.
24. Unity Real-Time Development Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://unity3d.com/unity>.
25. Unity Real-Time Development Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://unity3d.com/unity>.
26. Ritchie P. Practical Microsoft Visual Studio 2015 / Peter Ritchie – 2016.
27. Skeet J. C# in Depth / Jon Skeet – 2014.
28. P. Milgram and A. F. Kishino, Taxonomy of Mixed Reality Visual Displays IEICE Transactions on Information and Systems. / P. Milgram and A. F. Kishino // E77-D(12), – 1994. – С. 1321– 1329.
29. Іванова А. Технології віртуальної і доповненої реальності: можливості та перепони використання. / Іванова А. // Стратегические решения и риск-менеджмент. — 2018. — Вып. 3 (108). – ISSN 2618-947X.
30. R. Azuma, A Survey of Augmented Reality Presence: Teleoperators and Virtual Environments / R. Azuma – 1997. – С. 355– 385.

31. Класифікація і перспективні напрямки використання технології доповненої реальності/ Яковлєв Б. С., Пустов С. І. // Новини Тульського державного університету. Технічні науки. – 2013.
32. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

# ДОДАТОК А

Підсистема інтеграції тривимірної моделі об'єкту в мобільну систему навігації

Апробація

УКР.НТУУ"КПІ ім. Ігоря Сікорського" \_ТЕФ\_АПЕПС\_TV42135\_19М

Аркушів 4

Київ — 2019



Було подано тези щодо розробленого програмного продукту до XVII міжнародної науково-практичної конференції «Сучасні проблеми наукового забезпечення енергетики», яка проходила в Києві 23-26 квітня 2019 року. Тези надруковані в томі №2 на сторінці 98. Текст доповіді наведений нижче.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної  
науково-практичної конференції  
молодих вчених та студентів  
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

Рисунок А.1 — Обкладинка тому №2 збірника тез

<i>МОСКАЛЕНКО Ю.В., аспірант</i>	
<b>Машинне навчання для розв'язання логічних головоломок.</b>	93
<i>БАРАНИЧЕНКО О.М., магістрант гр. ТВ-71мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
<b>Веб-середовище для моделювання процесів міжагентної взаємодії в мережах Smart Grid.</b>	94
<i>ШВАЙКА Д.А., магістрант гр. ТР-81мн</i>	
<i>Керівник - доц., к.ф.-м.н. Тарнавський Ю.А.</i>	
<b>Використання техніки Structure from Motion в системі навігації.</b>	95
<i>ХАРАБАР В.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
<b>Система оцінювання екологічних збитків у мережі АЗС.</b>	96
<i>ОЛЕКСІЙ А.О., магістрант гр. ТВ-82</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
<b>Інтелектуальна система розпізнавання та передбачення намірів користувача.</b>	97
<i>МЕЛЬНИЧЕНКО А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
<b>Проблема формування схеми замкнутого простору у системах внутрішньої навігації.</b>	98
<i>МАРУНЯ А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
<b>Застосування нейронних мереж в мобільних застосунках.</b>	99
<i>МАРИЧ Т.І., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
<b>Генерація елементів цифрового контенту на основі аналізу тексту.</b>	100
<i>КРЮЧКОВСЬКА А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
<b>Програмний інструментарій виокремлення заданих об'єктів на зображенні.</b>	101
<i>КРУТЛИК Д.С., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
<b>Система розпізнавання жестів рук для людино-машинної взаємодії.</b>	102
<i>КОНКІНА Н.С., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
<b>Проблема вибору раціонального методу позиціювання користувача для системи навігації.</b>	103
<i>ЗАРИЦЬКИЙ В.П., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
<b>Нейромережеве архітектурне рішення для обробки аудіосигналів.</b>	104
<i>ВИТВИЦЬКИЙ Д.А., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Мажара О.О.</i>	
<b>Побудова сучасного веб-серверу на основі безсерверних технологій.</b>	105
<i>БРУНЬКО П.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
<b>Сегментація бур'янів на зображеннях з відеокамери наземного робота.</b>	106
<i>СОФІСНКО А.Ю., студент гр. ТР-52</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
<b>Серверна частина системи функціонування реєстру інформаційних ресурсів.</b>	107
<i>СОЛОМКІН Д.Г., студент гр. ЗПІ-ЗП-63</i>	
<i>Керівник - ст.викл. Гайдаржи В.І.</i>	

## **ПРОБЛЕМА ФОРМУВАННЯ СХЕМИ ЗАМКНУТОГО ПРОСТОРУ У СИСТЕМАХ ВНУТРІШНЬОЇ НАВІГАЦІЇ**

При розробці системи внутрішньої навігації замкнутого простору (приміщення) постає проблема оцінювання універсальності системи, тобто чи зможе система адаптуватися, якщо буде використана в іншому замкнутому просторі. Побудова схеми приміщення за допомогою систем класу Compas чи AutoCAD є доволі часовитратної процедурою і потребує багато ресурсів та додаткової інформації, такої як попередні заміри розміру приміщення та інших.

Вирішенням проблеми може бути застосування методу SLAM (Simultaneous Localization And Mapping) – це метод одночасної навігації і побудови карти-схеми. Метод використовується роботами чи автономними засобами пересування, потребують лише камери для побудови карти-схеми на невідомій місцевості чи приміщенні або оновленні вже існуючої схеми, використовуючи методи виокремлення типових точок з характерними рисами. Якщо розпізнавання виконується в приміщенні – цими точками є кути, вивіски, написи, предмети, що мають характерні риси. Якщо ж на відкритій місцевості, наприклад, в полі – це можуть бути кущі, що знаходяться на певній відстані один від одного[1].

Метод SLAM є концепцією, що поєднує в собі два процеси, що не залежать один від одного, в цикл послідовних розрахунків. Концепцію методу можна пояснити на прикладі лабіринту та робота: у нас є невідомий лабіринт, в якому знаходиться робот, що має лише камеру і йому необхідно за допомогою здвигов зображення на карті побудувати повну карту лабіринту для того, щоб наступний, хто потрапить в лабіринт не блукав, а вже чітко міг побудувати маршрут виходу з лабіринту. На практиці це означає, що робот шукає відповідь на запитання «Де я?» і «Як виглядає світ навкруги?».

Метод SLAM можна описати як постійно повторювана послідовність кроків:

1. Сканування навколишнього середовища.
2. Визначення зміщення на основі поточного кадру з попереднім.
3. Виділення на поточному кадрі характерних міток і відображення їх як точок.
4. Зіставлення точок кадру з мітками, що були отримані при попередньому скануванні приміщення та оновлення на основі отриманої інформації положення на карті.
5. Перевірка на цикл – чи не проходимо ми повторно по тій же місцевості.

Такий алгоритм дозволяє формувати не лише карту місцевості, а й траєкторію переміщення, яку можна використовувати для побудови наступного маршруту. Постійне коригування карти при повторному проходженні необхідно для зменшення похибки визначених характерних точок. Тобто при першому проходженні мітки можуть бути встановлені на об'єкти, що рухаються і при наступному проходженні їх вже немає на мапі і їх можна відсіяти і завдяки чому залишити лише нерухомі мітки. Кожному виділеному об'єкту зіставляється опис, що дозволяє його ідентифікувати на кожному кадрі, на якому він існує.

Організувати процес побудови схеми для звичайного користувача нескладно достатньо лише камери телефону та організації обробки отриманих знімків методом SLAM в режимі реального часу.

Перелік посилань:

1. Soldatov D. Mono visual simultaneous localisation and mapping for autonomous robots on TRIK/D. Soldatov. – Saint-Petersburg, 2016.